Proceedings of the 11th International Conference on Educational Data Mining

Kristy Elizabeth Boyer, Michael Yudelson (Eds.)

ACTNEXT BY ACT

華中師範大學
CENTRAL CHINA NORMAL UNIVERSITY

松鼠AI

University at Buffalo
Computational and Data-enabled
Science and Engineering (CDSE)

International Conference on Educational Data Mining (EDM) 2018
Proceedings of the 11th International Conference on Educational Data Mining, Kristy Elizabeth Boyer and Michael Yudelson (Eds.).
Buffalo, NY July 16-20, 2018

Preface

The 11th International Conference on Educational Data Mining (EDM 2018) is held under the auspices of the International Educational Data Mining Society at the Templeton Landing in Buffalo, New York. The conference, held July 15th through 18th, 2018, follows ten previous editions (Wuhan 2017, Raleigh 2016, Madrid 2015, London 2014, Memphis 2013, Chania 2012, Eindhoven 2011, Pittsburgh 2010, Cordoba, 2009 and Montreal 2008).

This year's EDM conference was highly competitive, with 145 long and short paper submissions. Of these, 23 were accepted as full papers and 37 accepted as short papers. All total, the combined acceptance rate of long and short papers is 41%. The acceptance rate for long papers is 16%. An additional 32 papers were accepted to the poster track.

This year's conference features three invited talks: Tiffany Barnes, Professor at North Carolina State University in Raleigh, NC; Jodi Forlizzi, Geschke Director of the HCI Institute and Professor at Carnegie Mellon University; and Jim Larimore, Chief Officer of Center for Equity in Learning at ACT, Inc.

Together with the Journal of Educational Data Mining (JEDM), the EDM 2018 conference supports a JEDM Track that provides researchers a venue to deliver more substantial mature work than is possible in a conference proceeding and to present their work to a live audience. Three such papers are featured this year. The papers submitted to this track followed the JEDM peer review process.

The main conference invited contributions to an Industry Track in addition to the main track. The EDM 2018 Industry Track received ten submissions of which six were accepted, a tangible improvement over last year, with only four submissions total, all of which were accepted. This expansion of the industry track represents an intentional goal to better connect industry researchers with the academic research community.

The EDM conference continues its tradition of providing opportunities for young researchers to present their work and receive feedback from their peers and senior researchers. The doctoral consortium this year features 14 such presentations, more than double compared to the prior year. In addition to the main program, there are four workshops: Educational Data Mining in Computer Science Education (CSEDM), Proposal Policy & EDM: Norms, Risks, and Safeguards, replicate.education: A Workshop on Large Scale Education Replication, and Scientific Findings from the ASSISTments Longitudinal Data.

We thank the sponsors of EDM 2018 for their generous support: ACTNext, University at Buffalo, Central China Normal University, and YiXue Inc. We are also thankful to the senior program committee and regular program committee members and reviewers, without whose expert input this conference would not be possible. Finally, we thank the entire organizing team and all authors who submitted their work to EDM 2018.

Kristy Elizabeth Boyer
University of Florida
Program Co-Chair

Michael Yudelson
ACT, Inc.
Program Co-Chair

Alexander Nikolaev
University of New York at Buffalo
General Chair

Organization

General chair: Alexander Nikolaev (University of Buffalo)

Program chairs: Kristy Elizabeth Boyer (University of Florida) and Michael Yudelson (ACT, Inc.)

Workshop & Tutorial Chairs: Sharon Hsiao (Arizona State University) and Joseph Jay Williams (University of Singapore)

Industry Track Chairs: Mary Jean Blink (TutorGen, Inc.) and Scott McQuiggan (SAS)

Doctoral Consortium Chairs: Collin F. Lynch (North Carolina State University) and Neil Heffernan (Worcester Polytechnic Institute)

JEDM Track Chairs: Min Chi (North Carolina State University) and Irena Koprinska (University of Sydney)

Poster & Demo Track Chairs: Wookhee Min (North Carolina State University) and Jacob Whitehill (WPI)

Publicity/Social Media Chair: Martina Rau (University of Wisconsin Madison)

Sponsorship Chair: Nicholas Lane (University of Buffalo)

Web chair: Paul Salvador Inventado (California State University, Fullerton)

Officers of the International Educational Data Mining Society: Mykola Pechenizkiy (Eindhoven University of Technology, Netherlands), Mingyu Feng (WestEd)

IEDMS Board of Directors:

Rakesh Agrawal	Data Insights Laboratories, USA
Ryan Baker	University of Pennsylvania, USA
Tiffany Barnes	North Carolina State University
Michael Desmarais	Ecole Polytechnique de Montreal, Canada
Sidney D'Mello	University of Notre Dame, USA
Neil Heffernan	Worcester Polytechnic Institute, USA
John Stamper	Carnegie Mellon University, USA
Kalina Yacef	University of Sydney, Australia

Senior Program Committee

Roger Azevedo	University of Central Florida
Ryan Baker	University of Pennsylvania
Tiffany Barnes	North Carolina State University
Gautam Biswas	Vanderbilt University
Michel Desmarais	Ecole Polytechnique de Montreal
Stephen Fancsali	Carnegie Learning, Inc.
Mingyu Feng	WestEd
April Galyardt	Carnegie Mellon University
Dragan Gasevic	Monash University
José González-Brenes	Chegg
Neil Heffernan	Worcester Polytechnic Institute
Sharon Hsiao	Arizona State University
Kenneth Koedinger	Carnegie Mellon University
James Lester	North Carolina State University
Noboru Matsuda	Texas A&M University
Agathe Merceron	Beuth University of Applied Sciences Berlin
Roger Nkambou	Université du Québec À Montréal (UQAM)
Zach Pardos	University of California, Berkeley
Philip I. Pavlik Jr.	University of Memphis
Mykola Pechenizkiy	Eindhoven University of Technology
Radek Pelánek	Masaryk University Brno
David Pritchard	Massachusetts Institute of Technology
Steven Ritter	Carnegie Learning, Inc.
Ma. Mercedes T. Rodrigo	Ateneo de Manila University
Cristobal Romero	University of Cordoba
Vasile Rus	The University of Memphis
John Stamper	Carnegie Mellon University
Stefan Trausan-Matu	University Politehnica of Bucharest
Stephan Weibelzahl	Private University of Applied Sciences Göttingen
Kalina Yacef	University of Sydney

Program Committee

Mirjam Augstein	University of Applied Sciences Upper Austria
Costin Badica	University of Craiova
Mary Jean Blink	TutorGen, Inc.
Nigel Bosch	University of Illinois Urbana-Champaign
Jesus G. Boticario	UNED
François Bouchet	Sorbonne Université
Alex Bowers	Columbia University
Javier Bravo Agapito	Universidad a Distancia de Madrid (UDIMA)
Keith Brawner	United States Army Research Laboratory
Renza Campagni	Università degli Studi di Firenze
Alberto Cano	Virginia Commonwealth University
Ted Carmichael	UNC Charlotte and Tutor Gen, Inc.
Mehmet Celepkolu	University of Florida
Min Chi	BeiKaZhouLi
Mihaela Cocea	University of Portsmouth
Cynthia D'Angelo	SRI International
Shayan Doroudi	Carnegie Mellon University
Bruno Emond	National Research Council Canada
Steve Fancsali	Carnegie Learning, Inc.
Jeremiah Folsom-Kovarik	Soar Technology, Inc.
Davide Fossati	Emory University
Carlos García-Martínez	University of Córdoba
Ilya Goldin	2U, Inc.
Joseph Grafsgaard	University of Colorado Boulder
Philip Guo	University of California San Diego
Jiangang Hao	Educational Testing Service
Vladimir Ivančević	University of Novi Sad, Faculty of Technical Sciences
Yang Jiang	Columbia University
Farzaneh Khoshnevisan	North Carolina State University
Irena Koprinska	University of Sydney
Sotiris Kotsiantis	University of Patras
Mayank Kulkarni	University of Florida
Sébastien Lallé	The University of British Columbia
Andrew Lan	Princeton University
Charles Lang	Columbia University
Sunbok Lee	University of Houston
Young-Jin Lee	University of Kansas
Chen Lin	North Carolina State University
Ran Liu	MARi, LLC
Vanda Luengo	Sorbonne Université
Ivan Luković	University of Novi Sad
Ling Luo	CSIRO

Maria Luque	University of Cordoba
Collin F. Lynch	North Carolina State University
Christopher Maclellan	Soar Technology, Inc.
Patricia Mahabir	MREI Inc.
Jessica McBroom	University of Sydney
Daniel Mccoy	Knowledge Net Consulting, LLC
Victor Menendez	Universidad Autónoma de Yucatán
Donatella Merlini	Università di Firenze
Cristian Mihaescu	University of Craiova
Wookhee Min	North Carolina State University
Piotr Mitros	
Carlos Monroy	Rice University
Bradford Mott	North Carolina State University
Andrew Olney	University of Memphis
Shai Olsher	University of Haifa
Luc Paquette	University of Illinois at Urbana-Champaign
Abelardo Pardo	University of South Australia
Lydia Pezzullo	University of Florida
Niels Pinkwart	Humboldt-Universität zu Berlin
Paul Stefan Popescu	University of Craiova
Thomas Price	North Carolina State University
Martina Rau	University of Wisconsin
Stu Rodgers	Tier1
Fernando Rodríguez	University of Florida
José Raúl Romero	University of Cordoba
Shaghayegh Sahebi	University at Albany - SUNY
Maria Ofelia "Sweet" San Pedro	ACT, Inc.
Olga C. Santos	aDeNu Research Group (UNED)
Margie Serrato	TutorGen, Inc.
Shitian Shen	North Carolina State University
Roi Shillo	Ben Gurion University
John Stamper	Carnegie Mellon University
Jessica Stokes	Center for Applied Learning Science at WGU
Jun-Ming Su	National University of Tainan
Ling Tan	Australian Council for Educational Research
Jennifer Tsan	North Carolina State University
Sebastián Ventura	University of Cordoba
Lucian Vintan/Lucian Blaga	University of Sibiu
Jacob Whitehill	Worcester Polytechnic Institute
Eric Wiebe	North Carolina State University
Joseph Wiggins	University of Florida
Fridolin Wild	Oxford Brookes University
Jamieka Wilkinson	University of Florida
Amelia Zafra Gómez	Department of Computer Sciences and Numerical Analysis

Alfredo Zapata González	Universidad Autonoma de Yucatan
Diego Zapata-Rivera	Educational Testing Service
Guojing Zhou	North Carolina State University

Sponsors

ACTNEXT™
BY ACT

華中師範大學
CENTRAL CHINA NORMAL UNIVERSITY

松鼠AI

UB
University at Buffalo
Computational and Data-enabled
Science and Engineering (CDSE)

Best Paper Selection

The two program chairs selected five best paper nominees based on the reviews and meta-reviews for each of those papers. The nominees were then sent to the members of the best paper awards committee. Each committee member read and ranked each one of the nominees. Ranking was compiled and the Best Paper Award was attributed to the most highly ranked paper. Next, the Best Student Paper award was attributed to the most highly ranked remaining paper that was also authored by a student.

Best Paper Award Committee:
Michel Desmarais
Tiffany Barnes
Roger Azevedo
Agathe Merceron
Kalina Yacef

Best Paper Nominees:

Predicting Quitting in Students Playing a Learning Game. Shamya Karumbaiah, Ryan S Baker, Valerie Shute

An Open Vocabulary Approach for Estimating Teacher Use of Authentic Questions in Classroom Discourse. Connor Cook, Andrew Olney, Sean Kelly, Sidney D'Mello

Impact of Corpus Size and Dimensionality of LSA Spaces from Wikipedia Articles on AutoTutor Answer Evaluation. Zhiqiang Cai, Art Graesser, Leah Windsor, Qinyu Cheng, David Shaffer, Xiangen Hu

Studying Affect Dynamics and Chronometry Using Sensor-Free Detectors. Anthony F. Botelho, Ryan Baker, Jaclyn Ocumpaugh, Neil Heffernan

Understanding Student Procrastination via Mixture Models. Jihyun Park, Renzhe Yu, Fernando Rodriguez, Rachel Baker, Padhraic Smyth, Mark Warschauer

Table of Contents

Full Papers

Short Papers

Posters

Industry Track Papers

Document Chunking and Learning Objective Generation for Instruction Design

Khoi-Nguyen Tran
IBM Research
Australia
khndtran@au1.ibm.com

Jey Han Lau
IBM Research
Australia
jeyhan.lau@au1.ibm.com

Danish Contractor
IBM Research
India
dcontrac@in.ibm.com

Utkarsh Gupta[*]
IBM Research
India
utgupta3@in.ibm.com

Bikram Sengupta
IBM Research
India
bsengupt@in.ibm.com

Christopher J. Butler
IBM Research
Australia
chris.butler@au1.ibm.com

Mukesh Mohania
IBM Research
Australia
mukeshm@au1.ibm.com

ABSTRACT

Instructional Systems Design is the practice of creating of instructional experiences that make the acquisition of knowledge and skill more efficient, effective, and appealing [18]. Specifically in designing courses, an hour of training material can require between 30 to 500 hours of effort in sourcing and organizing reference data for use in just the preparation of course material. In this paper, we present the first system of its kind that helps reduce the effort associated with sourcing reference material and course creation. We present algorithms for document chunking and automatic generation of learning objectives from content, creating descriptive content metadata to improve content-discoverability. Unlike existing methods, the learning objectives generated by our system incorporate pedagogically motivated Bloom's verbs. We demonstrate the usefulness of our methods using real world data from the banking industry and through a live deployment at a large pharmaceutical company.

1. INTRODUCTION

Recent estimates suggest that on average, an organization spends nearly $1200 per year, per employee for training.[1] Apart from the costs incurred in delivering training, significant costs are associated with instruction design activities such as sourcing and preparation of course materials. Currently, most of these activities are very human-intensive in nature, and they rely on the experience and expertise levels of instruction designers and intense reviews by subject-matter experts (SMEs) to achieve acceptable quality levels.

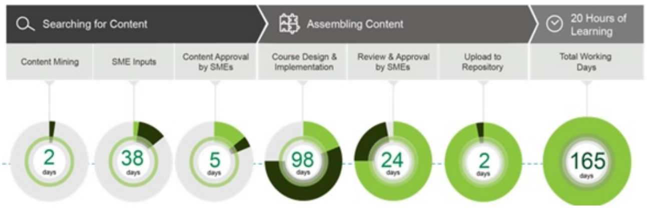


Figure 1: Typical course creation workflow

1.1 Course Creation: Workflow and Challenges

Figure 1 shows the typical steps involved in creating a new course. In the first step, instructional designers search for existing learning content that can be used for reference while developing the course. The learning objectives of the new (to be designed) course informs this search process. Reference materials may include existing courses and resources as well as other informal learning materials, such as those available in the form of media articles, blogs etc.

In the next step, the new course is designed and implemented by: extracting the relevant parts of the selected reference content, transforming them appropriately, and combining with newly developed materials to meet the overall training objectives. The new course content is finalized with SME review and approval. Finally, the course is uploaded to a repository for access by end users such as instructors and employees.

The average time taken to produce an hour of material this way can vary between 50 to 300 hours depending on the nature of the course being created.[2] The efficiency with which a new course can be assembled rests on two critical factors: (a) the ability to quickly locate an existing reference material, which is relevant to a learning objective that is part of the planned new course; and (b) the ability to identify (and eventually extract) appropriate parts of this material for use within the new course.

[*] Utkarsh carried out this work during his employment with IBM Research.

[1] `https://www.td.org/Publications/Magazines/TD/TD-Archive/2014/11/2014-State-of-the-Industry-Report-Spending-on-Employee-Training-Remains-a-Priority.`

[2] `https://www.td.org/Publications/Newsletters/Learning-Circuits/Learning-Circuits-Archives/2009/08/Time-to-Develop-One-Hour-of-Training.`

1.2 Contributions

In this paper, we present the first system of its kind that helps reduce the effort associated with sourcing reference material and course creation. We present algorithms for document chunking and automatically generating learning objectives from content as well as creating descriptive content meta-data that improves content-discoverability. Our novel methods for document chunking incorporate syntactic and stylistic features from text as well as a semantic vector-based representation of document text to identify meaningful chunks. Each chunk is physically persisted and a learning objective consisting of Bloom's verb [3] along with a descriptive keyphrase is generated and associated with each chunk. To the best of our knowledge, we are the first to generate learning objectives incorporating Bloom's verbs and our system is the first of its kind that directly addresses the challenges in instruction design.

We describe experiments using real-world data from two industries: banking and pharmaceutical. Our results on data from the banking industry shows that our document chunking methods are useful for instruction designers. We report an average user rating of 2 out of 3 in a blind study to assess the quality of chunks and an $F1$ score of 0.62 computed against expert generated gold standard chunks. Furthermore, in the challenging problem of generating learning objectives, the output from our system has an $F1$ score of 0.70 for predicting Bloom's verbs with an average user rating of 2.2 (out of 3) for the associated keyphrase. We also present details of a live deployment of our solution at a large pharmaceutical company.

2. RELATED WORK

To the best of our knowledge, our system is the first (commercial or prototype) that can automatically chunk/segment[3] learning material and label them with system-generated course objectives. We highlight some related work directly relevant to the subcomponents of document chunking and learning objective generation.

Document chunking: Broadly, most methods for chunking/segmentation of text rely on detecting changes in vocabulary usage patterns [11, 14, 15], identifying topical shifts [6, 7, 23], or employing graph based techniques to identify boundaries [9, 28]. The TextTiling [11] document segmentation algorithm uses shifts in vocabulary patterns to mark segment boundaries. Works such as Riedl and Biemann [25] adapt the TextTiling algorithm to work on topics generated by Latent Dirichlet Allocation. Glavis et al.[9] use a graph based representation of documents based on semantic relatedness of sentences to identify document segments. More recent work [1, 2] uses semantic distance computed based on vector embeddings to identify chunk/segment boundaries. Our work on document chunking is based on this direction of research. We use file format specific APIs to physically persist document chunks, retaining any stylistic and presentation elements from the original document.

Learning Objective generation: Most learning management solutions either rely on user provided learning objectives or automated methods to label documents with *existing* learning objectives specified in curricula [4]. Methods such as Bhartiya et al. [2] and Contractor et al. [5] use a curriculum hierarchy to label learning material with learning objectives. Milli and Hearst [22] simplify the problem of generating course objectives by directly using document keyphrases as learning objectives. Similarly, Lang et al. [16] and Rouly et al. [26] simplify generating objectives using topic modeling to identify candidate learning objectives, where Lang et al. [16] also suggest a system to match topics with Bloom's verbs. In contrast, we associate keyphrases with Bloom's verbs [3] and rerank them to select the best candidates for use as learning objectives. To the best of our knowledge, we are the first to *generate* pedagogically motivated learning objectives incorporating Bloom's verbs.

3. DOCUMENT CHUNKING

Course materials can often be very large and monolithic, covering a great number of topics and learning objectives, which makes consumption difficult. To make these course materials more discoverable, we automatically segment courses into smaller chunks that can persist independently in the course repository. We present three chunking approaches in the following sections.

3.1 Structure guided (SYNTACTIC-CHUNKER)

Section headings are often the most natural chunk boundaries as they reflect the organization of content by the document creator. Formats such as Microsoft Word have an underlying XML structure that allows us to create these natural chunks easily. However, for PDF documents, there is no encoded document structure information, but we can recover the section titles by analyzing the font sizes of text. To build the SYNTACTIC-CHUNKER, we use a combination of Apache PDFBox[4] for PDF documents, Aspose APIs[5] for Microsoft Office documents and Apache Tika[6] for all other document formats.

Algorithm 1 details the syntactic chunking algorithm where we do not have markers for the section headings. The algorithm aims to find the font size of the largest heading in the document for chunking. The SYNTACTIC-CHUNKER first groups the lines in the document by their font size (sequentially). For each of these font groups, the algorithm gathers statistics on the chunks that would be created for each group's font size. The largest font size (i.e. the top most section titles) is then chosen from the groups that satisfies the heuristics given in the chunking hyperparameters. An example heuristic is whether the number of chunks created by this font size is between 3 and 20, which is the number of sections or subsections we expect a document or a chapter to contain on average. The significant heuristics/hyperparameters for this algorithm are given in Table 1.

Finally, the line indices marking the start of the section headings are recovered through the font groups created earlier. These starting line indices are then further processed in the main algorithm for creating the physical chunks or storing the metadata.

[3] We use the word "chunk" and "segment" interchangeably, though a document chunk further refers to a physical embodiment of a document segment

[4] https://pdfbox.apache.org/

[5] https://docs.aspose.com/dashboard.action

[6] https://tika.apache.org/

Algorithm 1: Syntactic chunking algorithm

```
Input   : A path to the document
Output: A list of indices to lines/pages in the document
        marking the start of a chunk
LoadParameters("syntactic")
pdf ← LoadDocument()
lineText ← ExtractOnEachLine("text", pdf)
lineFS ← ExtractOnEachLine("fontsize", pdf)
// Font groups are contiguous groups of lines.
fgs ← $[(i, k-1) \mid \text{lineFS}[i] = \text{lineFS}[j], i \leq j < k]$
// Create chunk statistics for each font group
for $i, j \in$ fgs.$length$, $i = j$ do
    while $\text{lineFS}[\text{fgs}[i]] \geq \text{lineFS}[\text{fgs}[j]]$ do
        $\text{cStats}[\text{lineFS}[\text{fgs}[i]]] \mathrel{+}=$ GetStats(fgs[$j$])
        $j \leftarrow j + 1$
    end
end
// Select candidates from heuristics
cs ← $[\text{fg} \mid$ Heuristics(fg, cStats[fg]), $\forall \text{fg} \in \text{fgs}]$
chunkingFontSize ← LargestFontSize(cs)
// Return the chunk start boundaries
chunkStartIndices ←
  $[\text{fg}.startIndex \mid \text{lineFS}[\text{fg}] = \text{chunkingFontSize}, \forall \text{fg} \in \text{fgs}]$
```

Hyperparameter	Value	Description
font_group_lines	[1,3]	Minimum and maximum number of consecutive lines (of the same font size) to collapse.
n_chunks	[3, 20]	Minimum and maximum number of resulting chunks for each font size.
min_section_title_length	2	Minimum number of characters for a chunk's starting line.

Table 1: Syntactic-chunker hyperparameters.

Hyperparameter	Value	Description
min_par_to_stop	80	Threshold for the minimum number of lines to stop chunking.
trim_par	4	Proportion of starting and ending lines to ignore when searching for a chunk boundary.
word2vec_model	enwiki	Pre-trained WORD2VEC model.
max_vocab	1000	Number of most frequent word types to include from pre-trained WORD2VEC model.

Table 2: Semantic-chunker hyperparameters.

Algorithm 2: Semantic/hybrid chunking algorithm

```
Input   : A path to the document
Output: A list of indices to lines/pages in the document
        marking the start of a chunk
LoadParameters("semantic"/"hybrid")
pdf ← LoadDocument()
lineText ← ExtractOnEachLine("text", pdf)
// Vectorize words using pre-trained word vectors
lineVectors ← Vectorize(lineText)
/* Modifications for the hybrid algorithm                    */
lineFS ← ExtractOnEachLine("fontsize", pdf)
// Create font groups.
fgs ← $[(i, k-1) \mid \text{lineFS}[i] \equiv \text{lineFS}[j], i \leq j < k]$
// Vectorize the font groups
fgsV ← [VectorSum(Vectorize($\forall$lineText $\in fg$)) | $\forall$fg $\in$ fgs]
// Similar logic to the semantic algorithm
lineVectors ← fgsV
// Return the chunk start boundaries (function below)
chunkStartIndices ← FindSegments(lineVectors, startIndex)
/* Divide and conquer strategy                                */
Function FindSegments(lineVectors, startIndex):
    n ← Size(lineVectors)
    // Create the search area with the
       numParagraphsInChunk hyperparameter
    x ← n/numParagraphsInChunk
    y ← n/(1 − (1/numParagraphsInChunk))
    bestIndex ← $(x + y)/2$
    bestScore ← 1.0
    sumTop ← VectorSum(lineVectors[$1, x$])
    sumBot ← VectorSum(lineVectors[$x + 1$, n])
    for $x \leq i < y$ do
        sumTop ← VectorSum(sumTop, lineVectors[$i$])
        sumBot ← VectorSubtract(sumBot, n)
        cos ← Cosine($sumTop$, $sumBot$)
        if cos < bestScore then
            bestIndex ← $i$
            bestScore ← cos
        end
        chunkIndices.append([bestIndex + startIndex ])
        topVectors ← lineVectors[1, bestIndex]
        botVectors ← lineVectors[bestIndex + 1, n]
        // Hyperparameter minNumberOfLines as the
           stopping condition
        if Size(topVectors) > minNumberOfLines then
            chunkIndices.appendAll(FindSegments(topVectors,
             startIndex))
        end
        if Size(botVectors) > minNumberOfLines then
            chunkIndices.appendAll(FindSegments(botVectors,
             bestIndex + startIndex))
        end
    end
    return chunkIndices
```

3.2 Topically guided (SEMANTIC-CHUNKER)

Some document styles have ambiguous semantic separation of content, such as presentation slides, informal articles, and blogs. These document styles often have repeated font sizes and text that do not provide distinguishing characteristics for syntactic chunking. For example, presentation slides often have repeated font sizes for slide titles, causing the SYNTACTIC-CHUNKER to create a separate chunk for each slide. For these documents, their text content is more useful for inferring chunk boundaries than syntactic markers.

To chunk these documents, we use a divide-and-conquer approach based on topical or content shifts. We represent the content using mean bag-of-word embeddings, which are pre-trained WORD2VEC embeddings [20, 21].[7] We tokenise words using whitespace, and discard common symbols such as com-

[7] Word embeddings are trained on English Wikipedia.

mas and periods. When computing the mean embedding, stopwords are excluded.[8] The divide-and-conquer method first identifies a boundary that separates a document into two partitions that have the maximum cosine distance using the vector embeddings (providing topical diversity), and then recursively creates subpartitions until a minimum text length is reached. The search strategy is simpler compared to dynamic programming and iterative improvement techniques typically used in the literature [1] but we found this divide-and-conquer strategy performs encouragingly.

The pseudocode and hyperparameters for the SEMANTIC-CHUNKER algorithm with modifications to create the HYBRID-CHUNKER are in Algorithm 2 and Table 2, respectively. Both algorithms share similar hyperparameters and similar divide-and-conquer logic but on different data structures.

3.3 Hybrid method (HYBRID-CHUNKER)

The SEMANTIC-CHUNKER relies purely on content information for chunking, ignoring potentially usable structural information. From preliminary experiments, we observed that the SEMANTIC-CHUNKER occasionally partitions documents at arbitrary positions in the text. For example, a few lines after the start of a new section where the topical shift should be stronger. To resolve this, we developed a hybrid method that uses both structural and content information. Similar to the SYNTACTIC-CHUNKER, we record font sizes for each line, and gather lines that share a similar font size into a data structure. With these data structures, we apply the same divide-and-conquer approach used in the SEMANTIC-CHUNKER to recursively partition the document into multiple chunks. This forces the chunker to create partitions at natural text boundaries, when this information is available.

4. LEARNING OBJECTIVE GENERATION

Traditionally, learning objectives associated with courses are generated manually and are presented in a sentence-like structure. An example from a K-12 Science curriculum in the US: *Conduct an investigation to determine whether the mixing of two or more substances results in new substances.*[9]

Automatically generating these objectives can be posed as summarization problem where the task is to identify the "*learning skill*" imparted by the document. However, inferring a skill requires an in-depth understanding of the concepts presented, how they relate with each other, and in courses–such as those that teach soft-skills or behavioural skills–the relationships may be more abstract. Thus, in order to generate tractable yet usable learning objectives, we generate short sentences that are prefixed by a verb from the Bloom's taxonomy followed by a keyphrase. Recent work such as Milli and Hearst [22] contends with simply using keyphrases as learning objectives.

4.1 Candidate Keyphrase Selection

Existing methods for keyphrase extraction use a variety of different approaches. Some methods rely on supervision to extract keyphrases [13, 27, 29], while unsupervised methods often rely on graph-based ranking [19] or topic-based clustering [10, 17]. For our work, we rely on an accessible and effective keyphrase extraction method: IBM Watson Natural Language Understanding (NLU)[10] to extract keyphrases. NLU is one of many commercially available general purpose keyphrase extraction methods that performs effectively in general keyphrase extraction tasks [8, 12]. We also evaluated other methods such as a variant of TextRank [19], which has been used in extracting keyphrases from education material [4]. We chose NLU for the rest of this paper after a blind user study on 243 document chunks indicated a strong preference for these keyphrases as compared to the method employed by Contractor et al. [4]. Table 3 shows the proportion of useful keyphrases[11] for two keyphrase extraction methods. Further details and results are given in Section 5.3.

Method	% Useful Keyphrases
WATSON NLU	66
MODIFIED TEXTRANK [4]	51

Table 3: Percentage proportion of keyphrases identified by instructional designers as being "*useful*" for possible inclusion in learning objectives

As seen from Table 3, not all keyphrases extracted are useful for inclusion in learning objectives. Thus, to select candidate keyphrases for learning objectives from a general keyphrase list, we rank and select them using a combination of factors:

1. **Keyphrase score (α):** A score between 0-1 returned by the NLU indicating the importance of a keyphrase (1 = most important).
2. **N-gram TF-IDF score (β):** We compute an N-gram level TF-IDF score for each keyphrase using a large domain specific background corpus for IDF score computation.
3. **Inverse chunk frequency (γ):** We compute a chunk-level modified IDF score for each keyphrase where the IDF score is computed at the keyphrase level using sibling chunks of a given chunk.
4. **Google N-gram score (ϕ):** The Google Books N-gram service[12] returns the log-likelihood of a given N-gram from a language model trained on the Google Books corpus. We use the (normalized) rank for a keyphrase within a chunk as the N-gram score.
5. **Word token level overlap with document section titles (θ):** Tokens in a section title are likely to contain mentions of important concepts and this acts as a useful signal for selecting keyphrases for learning objectives.

[8] We use mallet's stopword list: https://github.com/mimno/Mallet/blob/master/stoplists/en.txt

[9] Sources: https://www.cs.ox.ac.uk/teaching/courses/2015-2016/ml/, https://www.nextgenscience.org/topic-arrangement/5structure-and-properties-matter.

[10] https://natural-language-understanding-demo.mybluemix.net/

[11] "Usefulness" is defined in terms of possible inclusion of a keyphrase in a learning objective, and not in terms of the "quality" of a keyphrase in a general keyphrase extraction task.

[12] https://books.google.com/ngrams.

	α	β	γ	ϕ	θ
bank	0	0.5	0	0.5	0
pharma	0.26	0.32	0	0.32	0.1

Table 4: Hyperparameter values for bank and pharma data for keyphrase re-ranking: α: orignal keyphrase score, β: N-gram TF-IDF score, γ: Inverse Chunk Frequency, ϕ: Google N-gram score, θ: Overlap with words in section titles.

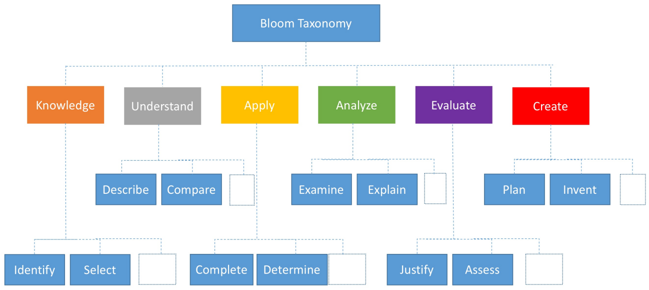


Figure 2: A representative taxonomy of Bloom's verbs

Let weight w_i be associated with each scoring factor f_i, where there are N factors. The weights of each factor is normalized to sum to 1.0 (i.e. $\sum_{i=0}^{N} w_i = 1.0$). Let $K_s^{(j)}$ denote the set of top-k keyphrases selected by the system for the j-th chunk (based on decreasing order of the score $\sum_{i=0}^{N} w_i f_i$). Let the average user rating (see Section 5.3) associated with the keyphrase set $K_s^{(j)}$ be denoted by s_j. Our goal is to select values of w_i that maximises s_j for all training examples:

$$max \frac{\sum_{j=1}^{M} s_j}{M} \tag{1}$$

where M is the number of training examples. The values for k and parameters w_i are estimated using grid search.

The tuned hyperparameters for the keyphrase selection are given in Table 4. We found that γ is not useful in these data sets, but maybe useful in other document collections where learning objectives are derived from a few chunks.

4.2 Bloom's Verbs Association

Bloom [3] proposes a taxonomy for promoting learning instead of rote memorization. Bloom's taxonomy aims to capture the whole pedagogy of learning, teaching, and processing information in a list of "action" verbs. These verbs (referred to as Bloom's verbs) characterize the activity involved in learning concepts.

Figure 2 shows a representative view of Bloom's taxonomy. For example, the verb *knowledge* has a list of child verbs such as *identify* and *select*. Similarly, other top-level verbs have their own set of verbs. We experiment with a subset of 10 verbs, as recommended by SMEs. We also explore another more condensed list as suggested by the same SMEs to investigate the potential of hierarchical options. We collapse the 10 verbs belonging to the same parent, resulting in 4 higher-level verb classes in Bloom's taxonomy. The verb classes used in our experiments are given in Table 5.

Original List	Collapsed List	Distribution	
		bank	pharma
identify	knowledge	542	323
define		85	12
recall		36	11
recognize		35	31
select		6	1
list		1	8
describe	understand	144	166
explain		127	65
outline	analyze	11	40
determine	apply	5	5

Table 5: Bloom's verbs used for generating Learning Objectives and their distribution from a random sample of 100 chunks. Each chunk often has more than one keyphrase describing it, requiring the SMEs to suggest a matching Bloom's verb.

To associate a verb from Bloom's taxonomy with a keyphrase learning objective, we train a multilayer perceptron (MLP) to predict a verb given a document (or chunk) and a candidate keyphrase. Thus, the MLP consists of two fully connected (dense) layers with ReLU activation functions[24] in each node. The input of the network is the mean bag-of-words embedding of the document text and the keyphrase.

Word embeddings are pre-trained WORD2VEC embeddings [20, 21] trained on the English Wikipedia. Word embeddings are kept static and not updated during back-propagation.[13] This approach of predicting bloom verbs was found to be very effective as shown in Section 5.3.

Two examples of generating learning objectives are shown in Table 6. They show the pairing of a Bloom's verb with various keyphrases. These pairings are presented to SMEs to evaluate, where their ratings allow us to determine the final rankings to select the most appropriate candidates as learning objectives for a piece of text. Note that the text in the examples (from a document chunk) has been truncated for presentation.

5. EXPERIMENTS

5.1 Data sets

We evaluate our chunking and learning objective systems on real-world documents from two industries: banking and finance (henceforth bank) and pharmaceuticals (henceforth pharma). Table 7 summarizes the word statistics of the two document collections used in our experiments.

The bank data set serves as our initial dataset for tuning and testing our methodology, which has a mix of 15 "formal" (e.g. Microsoft Word style) documents and 15 "informal" (e.g. HTML, MediaWiki style, Microsoft PowerPoint slides) documents.

The pharma data is a set of client-provided documents with a

[13] We also experimented with updating the embeddings (Facebook's fastText), but found little improvement and thus chose the simpler static model with fewer parameters.

Bloom's Verb	**Keyphrase**	**Avg. Rating**
describe	ach payments	3
explain	ach transaction flow	2.5
describe	ACH transactions	2.5
identify	ACH network	2
identify	ACH networks	2
identify	ACH payment request	2
describe	ACH payments industry	2
explain	internal ach transaction	2
identify	traditional ACH payments	2
identify	ACH	1
Text		
ACH Payments In this section we are going to take a look at a payment type generically known as small value electronic credit transfers, although they are referred to with a number of different names, including automated clearing house or ACH transactions, automatic clearing payments, electronic clearing payments and giro payments. ...		

Bloom's Verb	**Keyphrase**	**Avg. Rating**
explain	consumer payments	3
define	Large value payments	2
describe	payments industry	2
define	Small value payments	2
identify	consumer bill payments	1.5
recall	consumer payments operations	1.5
identify	corporate-to-corporate payments	1.5
identify	interbank payments	1.5
explain	payments	1.5
identify	banks	1
Text		
Business Overview Why focus on consumer payments? There are two sides to this question. First, why do banks focus on consumer payments? There are several reasons: Banks cannot accept consumer deposits without providing payment services linked to those accounts. While consumer deposits have always been important, they have never been as important as they are today. ...		

Table 6: Examples of generating learning objectives and their average ratings from SMEs.

similar distinction of formal and informal documents. The `pharma` data set consists of 382 courses containing 408 documents, where most courses only have one document. We develop our methodology on the `bank` data set and pursue a deployment on the `pharma` data set (detailed in Section 6). The remainder of this section describes our experimental results on the `bank` data set.

5.2 Evaluation: Document Chunking

For tuning and evaluation, we require gold standard chunks for the `bank` documents. To this end, we ask SMEs to chunk[14] these documents manually, resulting in 243 chunks in total for the 30 documents. The documents were chunked by SMEs (with inter-annotator disagreements of the chunk boundaries resolved) based on their understanding of the subject from an instructional design perspective. The SMEs opted for page level chunks and thus we build our measure of quality at the page level.

To measure the quality of our system against SMEs, we compute the average F1 score on their list of chunk boundaries. We omit the first chunk boundary as it always starts at page 1, and penalise duplicate page numbers (i.e. multiple sections on the same page). To illustrate the evaluation method, we give an example:

$$\text{system chunks} = [1, 4, 4]$$
$$\text{human chunks} = [1, 3, 4]$$

where each number in the list denotes the starting page number of a chunk. We omit the first chunk, yielding:

$$\text{system chunks} = [4, 4]$$
$$\text{human chunks} = [3, 4]$$

	bank	**pharma**
No. Documents	30	408
No. Word Tokens	376,570	1,251,712
Vocabulary Size	32,598	92,890

Table 7: Data set statistics.

Precision of the system is therefore 1/2 = 0.5 (the second starting page number "4" is penalised), the recall is 1/2 = 0.5, and thus F1 = 0.5.

There are a number of hyper-parameters for our chunking methods, which are available in Tables 1 and 2. We tune them manually based on the F1 score using a small labeled development set. Given the tuned models, we apply them to the `bank` documents.

From the chunking performance in Table 8, we found that for formal documents, the SYNTACTIC-CHUNKER (relying on the font size to detect natural chunk boundaries) has the highest accuracy for formal content. In contrast, for the informal content, where structural information may not be very indicative of natural chunk boundaries, we find that the SEMANTIC-CHUNKER gives better results as expected.

In order to qualitatively assess the results of our systems, we also evaluate them with a blind user study. Two expert instructional designers were presented the output of chunks by different chunking algorithms in random order and without information on the underlying algorithm. Each designer was asked to rate a chunk output with 1 (poor), 2 (acceptable), and 3 (good) based on their quality and usefulness from an Instructional Design point of view. Due to complexity and unsupervised nature of the task, ratings above 1 are strongly encouraging.

As seen in Table 8, the average ratings for all our best systems is greater than 1.5 indicating our system generated chunks could be acceptable and useful for instructional designers. Furthermore, we find that the scores from the user study reinforce the assessment that formal content (with well structured natural chunk boundaries) are reliably chunked using the SYNTACTIC-CHUNKER algorithm while informal content is better chunked using the SEMANTIC-CHUNKER algorithm.

Surprisingly, we find that the HYBRID-CHUNKER chunking algorithm performs poorly on informal content compared to

[14] Chunks are contiguous breaks in the document, so chunk boundaries can be succinctly described and compared using the starting line/page number for each chunk.

System	Doc Type	F1	Avg. Rating
SYNTACTIC-CHUNKER	Formal	**0.62**	**2.17**
	Informal	0.31	2.00
	Combined	0.47	2.08
SEMANTIC-CHUNKER	Formal	0.08	1.36
	Informal	**0.20**	**1.67**
	Combined	0.14	1.51
HYBRID-CHUNKER	Formal	**0.21**	1.49
	Informal	0.05	**1.77**
	Combined	0.13	1.63

Table 8: Results for Document Chunking on the bank data set. Bold values indicate the best performance for that system.

the SEMANTIC-CHUNKER. However, the average user evaluation rating shows that the resulting chunks are highly acceptable, as expected from initial trials in designing this algorithm. Our inspection shows that increasingly the granularity from lines to font groups simply means the desired chunk boundaries are often missed (and they are near misses), and that fewer chunks are created. We reason that fewer chunks are favorable to users when the document does not have clear chunking boundaries because of simplicity. Furthermore, our F1-score measure is strict, meaning near misses for chunk boundaries are also heavily penalized, but the chunk boundaries of the HYBRID-CHUNKER algorithm may be acceptable to the user. We also experimented with alternative methods such as repositioning the chunk start indices from the SEMANTIC-CHUNKER to match boundaries given by the SYNTACTIC-CHUNKER, but the resulting chunks were not favored by the SMEs in initial trials.

Overall, the SYNTACTIC-CHUNKER performs well on both formal and informal documents for the bank data set. On inspection of the informal documents, some contain sufficient structure for the SYNTACTIC-CHUNKER to infer the desired chunking boundaries, whereas documents with non-usable structures, the SEMANTIC-CHUNKER provides more favorable chunking boundaries. We also reason that the higher ratings for the SYNTACTIC-CHUNKER is due to the SYNTACTIC-CHUNKER finding section headings for chunking boundaries, which seems to be preferred by users, whereas another grouping of pages for the chunk may be more appropriate. These chunking systems provide variety, ensuring that we have a suitable set of chunks for any document.

5.3 Evaluation: Learning Objective Generation

To collect annotation for evaluation and for training the Bloom's verb MLP and for keyphrase selection, we present to SMEs: a document chunk (manually chunked by different SMEs in Section 5.2) and the top-10 NLU generated keyphrases and ask them to (1) rate the keyphrase in terms of usefulness as a learning objective suffix on an ordinal scale from 1–3 (same as chunking evaluation) and (2) select an appropriate Bloom's verb (out of 10 verbs) for the particular keyphrase.

We randomly sample from the full 243 document chunks and collect annotations for 100 chunks, where each chunk is annotated by 2 SMEs. We aggregate these keyphrase ratings by taking the mean rating. For Bloom's verb selection, we ask the judges to agree on a particular verb if there is discrepancy. To generate gold standard for the condensed verbs (4 classes), we map the original 10 classes to the 4 classes, as given in Table 5.

	P@1	P@3	P@5
Avg. Rating	1.97	**2.23**	2.20
Precision	0.5	**0.5**	0.45

Table 9: bank: Candidate Keyphrase Selection for Learning Objective Generation

5.3.1 Candidate Keyphrase Selection

We use 10-fold cross-validation at the chunk level for our experiments. We select the top-k keyphrases for each chunk as candidates for the learning objectives of that chunk. From Equation 1, the tuning of factor weights is based on the average user rating of these top-k keyphrases.

We evaluate the quality of candidate keyphrase selection using the average user rating of the selected keyphrases, and Precision@N defined as

$$P@N = \frac{k_g \cap k_s}{|k_s|} \qquad (2)$$

where k_g is the set of gold standard keyphrases that have an average user rating of at least 1.5[15], and k_s is the set of top-k keyphrases selected by the system. This measure shows whether our selection methods are returning the keyphrases that are relevant for each chunk as determined by the SMEs.

From Table 9 our keyphrase selection method has a P@5 of 0.45 with a high average user rating. This means that 45% of the top 5 keyphrases selected contain the gold standard keyphrases.

5.3.2 Selecting Bloom's Verbs

Given a document and its verbs from the Bloom taxonomy, we train an MLP and optimise its hyperparameters based on 10-fold cross-validation at the chunk level. We use the evaluation metric of mean F1 score over the 10-folds.[16] We use 2 test sets: (1) all keyphrases and (2) top-5 keyphrases predicted by our system. Note that in each fold, the training data remains the same, but test set (2) is a subset of (1).

We present the classification performance of Bloom's verbs in Table 10. As expected, the performance in the 4-class prediction task is better than the 10-class prediction due to less confusion amongst classes. Baseline experiments where we assign the majority class for all predictions show a consistent 0.10 drop in F1-score for both the 4-class and 10-class prediction scores.

[15] We want our system to select only good quality keyphrases.

[16] For a particular fold, we compute weighted F1, where it is weighted by the number of true instances for each class.

Test Set	F1 4-Class	 10-Class
All KP	0.69	0.51
System Top-5 KP	0.70	0.53

Table 10: bank: Bloom's verb (BV) prediction performance. "KP" denotes keyphrase.

	P@1	P@3	P@5
Avg. Rating	1.24	1.35	**1.38**
Precision	0.1	0.3	**0.32**

Table 11: pharma: Candidate Keyphrase Selection for Learning Objective Generation

6. DEPLOYMENT

Making content discoverable is a key challenge faced by talent development teams in organizations worldwide. Our system addresses this challenge and is currently being piloted at one of the world's largest pharmaceutical companies to help organize their learning content.

Experiments and Tuning: Using the pharma data shared by the pharmaceutical company (statistics in Table 7), we repeated the bank data set experiments on this data. The pharmaceutical SMEs only wanted generation of document level learning objectives, and not document chunking. Thus, we describe only the experiments for this task. As with the bank experiments, we ask SMEs to rate predicted keyphrases and select the appropriate verb (from the Bloom's taxonomy) given a document[17] and keyphrase.[18] For learning objective keyphrase selection and Bloom's verb prediction, we train and tune the systems with 10-fold cross-validation as before.

Keyphrase selection and Bloom's verb prediction performance for pharma are presented in Table 11 and Table 12. We find that our candidate keyphrase average rating and precision is lower than what was seen for banking data. We hypothesize a reason for this is due to the extremely dense and domain specific content as well as the requirement of complete documents without chunking when generating learning objectives.

Furthermore, many documents from the pharmaceutical company refer to chemical compounds and chemical formulae, which resulted in skewed TF-IDF weights while selecting candidate keyphrases. Our hypothesis is also backed by the score weights for TF-IDF become less important for pharma data as compared to bank data. We note that the Google N-grams scores were useful for re-ranking keyphrases in both domains. The results also suggest that domain-specific adaption of keyphrase extraction methods (eg. supervised methods) may be required for learning objective generation in content that is very technical.

Test Set	F1 4-Class	 10-Class
All KP	0.66	0.50
Top-5 System KP	0.71	0.48

Table 12: pharma: Bloom verb prediction performance. "KP" denotes keyphrase.

System	Avg. Time Per Document (seconds) bank	 pharma
SYNTACTIC-CHUNKER	0.41	0.20
SEMANTIC-CHUNKER	0.40	0.20
HYBRID-CHUNKER	0.49	0.27
KEYPHRASE	0.02	0.02
KEYPHRASE RERANKING	0.03	0.02
BLOOM-VERB	0.05	0.04

Table 13: Throughput: Document Chunking, Keyphrase generation, candidate keyphrase selection, and bloom verb prediction (in seconds)

For Bloom's verb prediction (Table 12), we see a marginally lower performance, but the trend largely remains the same.

6.1 Commercial Deployment

A collection of over $20,000$ learning courses have been labeled with learning objectives generated by our system and are being imported into existing learning management systems used by the organization. This is to help the organization retrieve courses efficiently, identify similar course material and prioritize new course development as it allows them identify gaps in their course material by checking course objectives not covered existing in course material. We briefly describe the architecture of our full system as this is the eventual deployment goal.

6.2 System Architecture

Broadly, the system consists of three subsystems (see Figure 3): (1) **UI and Business logic layer**, which exposes interfaces for search and enforces business logic for user access; (2) **Data Analytics layer**, which are Web services for document chunking, keyphrase extraction, learning objective generation. Additional web services that generate different metadata can be easily plugged in and integrated into our system; and (3) **Data Storage and Search**, where we use Apache Solr to store all generated metadata and document text and to enable search. An illustration of the architecture is presented in Figure 3. Physical documents can either be stored locally or can be accessed via remote requests to learning management systems. Data ingestion from formal course repositories as well as informal sources (web based or Intranet) are supported.

We use document format specific APIs to physically persist document chunks in their original file formats. Our system exposes a simple search interface by which users can query the system using learning objectives. The system allows refinement of search results and also defines user workspaces where course packages can be created and shared.

[17] These were the original documents and were not chunked.

[18] We collect annotations for a random 25% subset of the 408 (original) documents, as SMEs simply did not have the time to evaluate all documents due to their length.

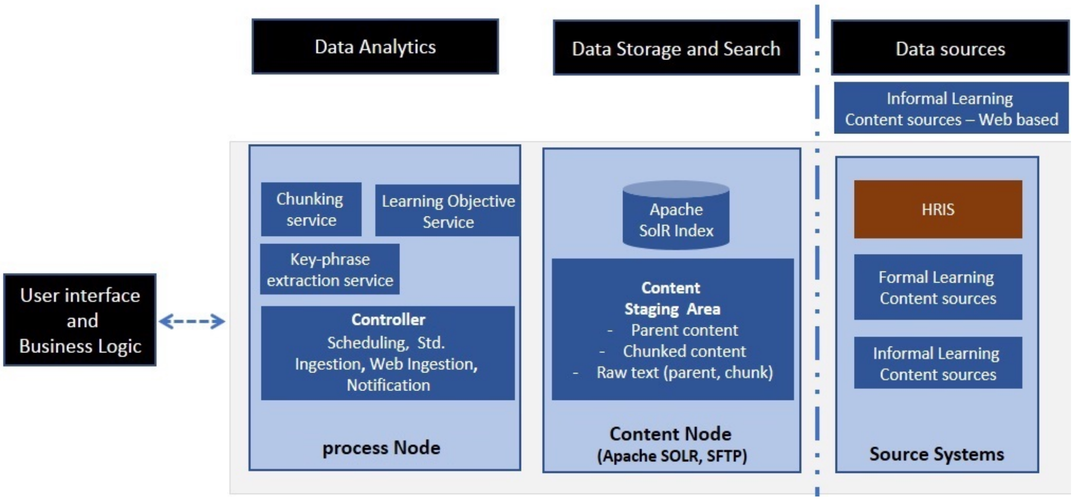

Data Analytics
Data Storage and Search
Data sources
Informal Learning Content sources – Web based
User interface and Business Logic
Chunking service
Learning Objective Service
Key-phrase extraction service
Controller
Scheduling, Std. Ingestion, Web Ingestion, Notification
process Node
Apache SolR Index
Content Staging Area
- Parent content
- Chunked content
- Raw text (parent, chunk)
Content Node (Apache SOLR, SFTP)
HRIS
Formal Learning Content sources
Informal Learning Content sources
Source Systems

Figure 3: High-level architecture diagram

Table 13 summarizes the average throughput for each of our components (computed on an Intel i5 6300 2.4 Ghz CPU with 8 GB RAM), demonstrating its speed and ease of scalability for large scale processing.

7. DISCUSSION AND CONCLUSION

In this paper, we presented the first system that automatically chunks learning material and generates learning objectives derived from content. It consists of modular subcomponents that require little training data for adaptation. The cloud based web service architecture enables effective use of each of its capabilities.

Our system uses a state-of-the-art embedding-based approach to chunk learning material into meaningful chunks. It also uses generic structural features from the document to guide chunking. It employs a novel methodology for generating learning objectives, which combines automatically generated verbs from Bloom's taxonomy and extracted keyphrases.

Our system's capabilities are being used by a large pharmaceutical company to organize learning material. We present detailed experiments on two different domains that demonstrate the applicability of our work.

In future work, we look to extend the work with improvements to our document ingestion capabilities, such as supporting images and videos using OCR and extracting headers and footers, and tabulated data. We would also like to add capabilities that aid instructional designers with other aspects of course design, such as discovering similar courses, summarizing documents, and improving learning objective generation to support a wider set of verbs from Bloom's taxonomy as well as supervised approaches for keyphrase generation in highly technical domains.

Acknowledgements

We thank our colleagues from IBM Global Business Services for leading this project on the business side. In particular, many thanks to Prasanna C Nair, Sandra Misiaszek, Madhusmita P Patil, Narasimhan K Iyengar, Renjith K Mathew, Partha S Guha, Pinaki Chakladar, Richa Sethi, Tulasi S Manepalli, Anindita Gupta, and Vinod Uniyal. Our research would not have been possible without their vision, support, data, expertise, and client engagements.

8. REFERENCES

[1] A. A. Alemi and P. Ginsparg, *Text segmentation based on semantic word embeddings*, arXiv preprint arXiv:1503.05543, (2015).

[2] D. Bhartiya, D. Contractor, S. Biswas, B. Sengupta, and M. K. Mohania, *Document segmentation for labeling with academic learning objectives*, in Proceedings of the 9th International Conference on Educational Data Mining (EDM), 2016, pp. 282–287.

[3] B. Bloom, D. Krathwohl, and B. Masia, *Bloom taxonomy of educational objectives*, Allyn and Bacon, 1984.

[4] D. Contractor, S. Negi, K. Popat, S. Ikbal, B. Prasad, S. Vedula, S. Kakaraparthy, B. Sengupta, and V. Kumar, *Smarter learning content management using the learning content hub*, IBM Journal of Research and Development, 59 (2015).

[5] D. Contractor, K. Popat, S. Ikbal, S. Negi, B. Sengupta, and M. K. Mohania, *Labeling educational content with academic learning standards*, in Proceedings of the 2015 SIAM International Conference on Data Mining (SDM), 2015, pp. 136–144.

[6] L. Du, W. L. Buntine, and M. Johnson, *Topic segmentation with a structured topic model*, in Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2013, pp. 190–200.

[7] L. Du, J. K. Pate, and M. Johnson, *Topic segmentation with an ordering-based topic model*, in Proceedings of the 29th AAAI Conference on Artificial Intelligence, 2015, pp. 2232–2238.

[8] A. Gangemi, *A comparison of knowledge extraction tools for the semantic web*, in Proceedings of the 10th Extended Semantic Web Conference (ESWC), 2013, pp. 351–366.

[9] G. Glavaš, F. Nanni, and S. P. Ponzetto, *Unsupervised text segmentation using semantic relatedness graphs*, in Proceedings of the 5th Joint Conference on Lexical and Computational Semantics, 2016.

[10] M. Grineva, M. Grinev, and D. Lizorkin, *Extracting key terms from noisy and multitheme documents*, in Proceedings of the 18th International Conference on World Wide Web, 2009, pp. 661–670.

[11] M. A. Hearst, *Texttiling: A quantitative approach to discourse segmentation*, tech. rep., University of California at Berkeley, 1993.

[12] L. Jean-Louis, A. Zouaq, M. Gagnon, and F. Ensan, *An assessment of online semantic annotators for the keyword extraction task*, in Proceedings of the 13th Pacific Rim International Conference on Artificial Intelligence, 2014, pp. 548–560.

[13] X. Jiang, Y. Hu, and H. Li, *A ranking approach to keyphrase extraction*, in Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2009, pp. 756–757.

[14] A. Kazantseva and S. Szpakowicz, *Topical segmentation: a study of human performance and a new measure of quality*, in Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2012, pp. 211–220.

[15] A. Kazantseva and S. Szpakowicz, *Measuring lexical cohesion: Beyond word repetition*, in Proceedings of the 25th International Conference on Computational Linguistics (COLING), 2014, pp. 476–485.

[16] C. Lang, R. Levy-Cohen, C. Woo, B. Roberts, S. Pepe, R. Verma, and Y. Xu, *Automated extraction of learning goals and objectives from syllabi using lda and neural nets*, Proceedings of the 8th International Conference on Learning Analytics and Knowledge, (2018).

[17] Z. Liu, P. Li, Y. Zheng, and Maosong, *Clustering to find exemplar terms for keyphrase extraction*, in Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, 2009, pp. 257–266.

[18] M. D. Merrill, L. Drake, M. J. Lacy, J. Pratt, and I. R. Group, *Reclaiming instructional design*, Educational Technology, (1996), pp. 5–7.

[19] R. Mihalcea and P. Tarau, *TextRank: Bringing order into texts*, in Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2004.

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space*, in arXiv:1301.3781, 2013.

[21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, *Distributed representations of words and phrases and their compositionality*, in Advances in Neural Information Processing Systems (NIPS), 2013, pp. 3111–3119.

[22] S. Milli and M. A. Hearst, *Augmenting course material with open access textbooks*, in Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications (BEA), 2016, pp. 229–234.

[23] 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 (CIKM), 2009, pp. 1553–1556.

[24] V. Nair and G. E. Hinton, *Rectified linear units improve restricted boltzmann machines*, in Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10, USA, 2010, Omnipress, pp. 807–814.

[25] M. Riedl and C. Biemann, *TopicTiling: a text segmentation algorithm based on LDA*, in Proceedings of ACL Student Research Workshop, 2012, pp. 37–42.

[26] J. M. Rouly, H. Rangwala, and A. Johri, *What are we teaching?: Automated evaluation of cs curricula content using topic modeling*, in Proceedings of the 11th International Computing Education Research, ICER '15, New York, NY, USA, 2015, ACM, pp. 189–197.

[27] M. Song, I.-Y. Song, , and X. Hu, *A flexible information gain-based keyphrase extraction system*, in Proceedings of the 5th ACM International Workshop on Web Information and Data Management, 2003, pp. 50–53.

[28] A. Tagarelli and G. Karypis, *A segment-based approach to clustering multi-topic documents*, Knowledge and Information Systems, 34 (2013), pp. 563–595.

[29] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning, *Kea: Practical automatic keyphrase extraction*, in Proceedings of the 4th ACM Conference on Digital Libraries, 1999, pp. 254–255.

Modeling Math Identity and Math Success through Sentiment Analysis and Linguistic Features

Scott Crossley
Georgia State University
Atlanta, GA 30303
scrossley@gsu.edu

Jaclyn Ocumpaugh
University of Pennsylvania
Philadelphia, PA 19127
jlocumpaugh@gmail.com

Matthew Labrum
Reasoning Mind
Houston, TX 77507
mlabrum@reasoningmind.org

Franklin Bradfield
Georgia Tech Research Institute
Atlanta, GA 30318
Franklin.Bradfield@gtri.gatech.edu

Mihai Dascalu
University of Bucharest
Bucharest, Romania 060042
mihai.dascalu@cs.pub.ro

Ryan S. Baker
University of Pennsylvania
Philadelphia, PA 19127
ryanshaunbaker@gmail.com

ABSTRACT

A number of studies have demonstrated strong links between students' language features (as found in spoken and written production) and their math performance. However, no studies have examined links between the students’ language features and measures of their Math Identity. This project extends prior studies that use natural language processing (NLP) features to examine student language features and math performance, replicating their analyses. The study then uses NLP features to model students’ Math Identity. Specifically, the study compares performance on basic math skills within an online math tutoring system to both student language (as captured in emails to a virtual pedagogical agent) and to survey measures of Math Identity (math self concept, interest, and value). Language features were analyzed by a number of NLP tools that extracted information related to text cohesion, lexical sophistication, and sentiment. The findings indicate weak to medium relationships between math scores and Math Identity and language features were able to predict a significant amount of the variance in each Math Identity variable and in math scores. The potential for these measures to inform interventions for students with lower Math Identity is discussed.

Keywords

Natural language processing (NLP), math, math identity, student success, on-line learning

1. INTRODUCTION

Educational Data Mining (EDM) has, among its many applications, been employed to better understand student-level differences that are important to personalization efforts in educational settings [1, 2]. These include efforts to better understand constructs like student engagement (e.g., [3]), self-efficacy [4], and self-concept [5]. Many of these studies have relied upon sensors (e.g. posture sensors, vocal recognition, heartbeat, video, sweat/skin conductance, EEG), which can sometimes make it challenging to implement interventions *in situ*. Research using student interaction data has become more common even when modeling highly qualitative constructs like student engagement (c.f., [3]), but to date, much of these efforts have focused on temporally short variables (e.g., state-based variables like behaviors and affect), rather than on trait-based variables such as identity, which are larger in scope and duration.

Work in related research areas has shown results that suggest that trait-based variables may be a promising area for investigation. Within the EDM community, there is now a growing body of research on identity-related constructs, such as motivation and self-regulated learning strategies (cf. [6]). Meanwhile, the related field of Natural Language Processing (NLP) has demonstrated relationships between language use and personality characteristics (cf.,[7, 8]). Detecting a construct like identity, which underlies motivation and goals [9], could further advance efforts toward personalized learning within educational setting, including the development of effective intervention strategies.

Identity, broadly, refers to a person’s sense of who they are and the development of an identity permits people to make predictions about their abilities to navigate different aspects of their life (cf. [9]). While identity is the focus of this study, we do not attempt to investigate all aspects of student identity, but instead focus specifically on how they identify with math. Math Identity is often described as “the association between math and the self” [10], a definition that might be paraphrased as the degree to which one considers oneself to be a math person. We do so within the context of Reasoning Mind, a blended learning curriculum that offers significant metacognitive support to K-6th grade students through an on-line learning platform [11]

Specifically, we use language features produced in within-system emails to predict three aspects of Math Identity in self-reported survey data: math self-concept, math interest, and math value. These constructs have been used to understand social influences on mathematic achievement in previous studies of identity (e.g., [12]). In addition, we examine links between math success in the system and the three Math Identity scales. We also use language features in the language produced by students to model math success, math value, math self-concept, and math interest. Our goal is to examine the potential for linguistic predictors within student data to identify math success and identity. If successful, such linguistic predictors could be used to better identify students in need of intervention.

2. Language and Math Ability

The body of research demonstrating connections between proficiency in language and math skills continues to grow, becoming more robust as researchers explore the potential underlying causes. Early studies focused on links between scores on math and language tests. For instance, [13] found that students who scored high on an algebra test also scored well on language tests. Using a more difficult algebra test produced a stronger relationship between algebraic notation and language ability. Similarly, [14] reported links between language and math skills, but also found that language skills differed in their degree of relation with math knowledge. For example, general verbal ability was indirectly related with symbolic number skills while phonological skills were directly related to arithmetic knowledge.

Other research has focused on more indirect links between math and language skills, such as reading ability. For example, Hernandez [15] found significant positive correlations between reading and math scores in standardized tests. Based on these findings, Hernandez recommended that reading skills and reading strategies should be factored into math instructions to increase math ability, especially for poor readers. In another study, LeFevre et al. [16] reported that language ability was positively related to number naming, but that non-language abilities such as quantitative skills and spatial attention were stronger predictors of math ability than language abilities.

A number of recent studies have begun to examine links between the language features found in students' language production and their success in math learning using NLP tools. For instance, Crossley et al. [17] examined linguistic and non-linguistic features of elementary student discourse while students were engaged in collaborative problem solving within an on-line math tutoring system. NLP tools that reported on affect, text cohesion, and lexical sophistication were used to extract linguistic information from transcribed student speech. These language features along with a variety of non-linguistic features such as gender, age, grade, and school were used to predict pre- and post-test math scores. The results showed that language features related to cohesion, affect, and lexical proficiency explained around 30% of the variance in students' math scores, while the selected non-language features were not significant predictors. A second study by Crossley and colleagues examined students' forum posts in an online tutoring system. Using these posts, Crossley et al. [18] investigated relationships between math success, click-stream data within the system, and language features reported by NLP tools for students in a university level blended math class (i.e., a class with both on-line and traditional face to face instruction). The study found that math success was best predicted by a non-language feature (days on the system) and language features related to affect (egotism), syntactic complexity and text cohesion. Specifically, more complex syntactic structures and fewer explicit cohesion devices equated to higher course performance. The linguistic model also indicated that less self-centered students and students using words related to tool use were more successful. In addition, the results indicated that students that are more active in on-line discussion forums are more likely to be successful. In a final study, Crossley and Kostyuk [19] examined links between the language features of young students' language production (grades 2nd through 5th) while e-mailing a virtual pedagogical agent in an online math tutoring system, and success within that system. Using NLP tools that reported language features related to affect, lexical sophistication, and text cohesion, Crossley and Kostyuk found that students who expressed more certainty in their writing and followed standardized language patterns scored higher in math assessments. In addition, students from higher grades who met more objectives, received more messages from teachers, and sent fewer messages to the agent, performed better on math problems.

Overall, these studies demonstrate that features from students' language productions can be used to predict math success (i.e., performance) in a variety of domains and across a number of ages and proficiency levels. In general, older students who produce more complex language, which is more positive and less self-centered, tend to have stronger math skills. For younger students, adherence to expected language patterns relates to higher math performance. However, to our knowledge, no research has attempted to extend this approach to predicting larger student identity features that are trait-based such as Math Identity.

3. Math Identity

Math Identity, or the degree to which one considers oneself a "math person," has become an area of interest among social scientists hoping to better understand what drives students to enter Science, Technology, Engineering, and Math (STEM) fields (cf. [20]). However, broader issues of self-definition (identity) are not new to educational research, especially when considering long-term development. For example, Bandura's research [21] on self-efficacy discusses the role of self-attributional processes (including a wide range of self-definitions studied by Bem, [22] many of which are directly related to educational identities. In this research, a student's cognitive appraisal (self-evaluation of ability) is thought to be susceptible to a form of confirmation bias where the student ignores demonstrable achievements and improvements when they contrast with a previously established self-definition [21]. Bandura's observations on the role of self-definitions in the development of self-efficacy are highly compatible with other research paradigms, which describe identity as an anchor that people use to understand their own interests and abilities [23]. This may explain Bandura's findings that students who show improvement that is contrary to self-appraisals often attribute their performance to environmental factors rather than to their own persistence [21].

Constructs considered to be a core part of one's identity are long thought to start developing in adolescence ([24]. There is some support that Math Identity should be included in this timeframe with research suggesting that it develops early in life. For instance, [25] showed that students who start in a non-STEM degree program rarely transfer into a STEM program (despite the high frequency of major changes more generally). Similarly, within the EDM community, student engagement indicators in middle school online mathematics tutors have been shown to correlate with college enrollment more generally [26], and with STEM-major enrollment more specifically [27]. Math Identity is most often studied through ethnographic studies (e.g., [28]), implicit association tests (e.g., [29, 10]), and surveys (e.g., [30, 31]).

In this study, we operationalize Math Identity as math self-concept, math interest, and math value. We defined these constructs using self-report scales adapted from Ryan & Ryan [12], who examined how these constructs performed during conditions likely to trigger stereotype threat effects. While these are well-established constructs in research on the effects of social evaluations of mathematics, they are not unique to research on identity. In addition to their appearance in Bandura's work, they appear in Eccles' [32] expectancy value theory, where self-efficacy (among a variety of other factors) is hypothesized to influence both intrinsic value (interest) and utility value (the usefulness of the task). We discuss each of these briefly below.

3.1.1 Math Self-Concept

Research in self-concept overlaps considerably with two related constructs—identity and self-efficacy—because all three are related to the mental schema a person uses when calculating their ability to negotiate different challenges in their lives. In general, social-psychologists are more likely to refer to the concept of identity when discussing issues related to social processes, while they are more likely to use the term self-concept when discussing internal mental processes ([9]).

In education research, self-concept and self-efficacy are often used to discuss domain-specific evaluations (e.g., self-concept in mathematics), and they are sometimes used synonymously. However, there are education researchers who draw a distinction between these two constructs, limiting the term self-efficacy to self-evaluations of specific tasks, often specifying that it must be measured directly after the task has been completed [33, 34]. For example, they might use a Likert scale administered after each math problem to measure self-efficacy by asking a student to indicate his/her confidence that each problem had been completed correctly.

In this research tradition, self-concept is a broader measure of ability within the domain, where its meaning more closely approaches its use among social-psychologists, who tend to define it as a theory of self (e.g., [35]) which often operates below the level of consciousness, guiding people's interpretations and expectations of external events (cf. [9]). For example, in a situation where a student failed a task in a domain for which they have high self-concept, they might be more willing to retry than someone with low self-concept. Alternatively, they might interpret the task as flawed since their performance did not match the expectations created by their self-concept.

Like researchers who study educational outcomes, social psychologists tend to believe that people develop self-concept from experience, so that those with more shallow or limited experiences are likely to be more susceptible to changes in self-concept [35]. For example, academic self-concept tends to be positively correlated with achievement indices, [36], but there appears to be some reciprocity in this relationship. High self-concept can make students more likely to persist through difficult mathematics instruction, leading to improved academic outcomes. However, repeated failure could theoretically lower self-concept, particularly if a student did not have other mastery experiences in mathematics to serve as a sort of buffer.

3.1.2 Interest in Mathematics

Motivational research defines interest as the propensity to engage with a particular subject over time through both affective and cognitive components [37]. Studies on the relationship of interest to other constructs such as self-concept have repeatedly found that self-concept drives intrinsic interest in a given subject [38, 39], with theorists suggesting that as self-efficacy increases, it becomes safe for the ego to become invested in a particular topic [40].

Researchers have identified a number of simple strategies that appear to increase interest in the classroom, such as creating more challenging tasks for students or adding variety to the ways in which a student is asked to perform a task. However, others caution that some of these strategies may only improve situational interest (e.g., [37]), suggesting that intrinsic interest (which they refer to as individual interest) is almost always self-driven, possibly because it seems to be fed by increased self-efficacy. Others researchers have found that interest is highly susceptible to contextual effects that vary from student to student (cf. [39]). Researchers in Career Theory (e.g., [41]) have found that interest, like self-efficacy, is directly responsive to performance success and failure.

Interest is an important complement to self-concept when defining Math Identity, since its development is known to improve self-regulatory strategies [37]. Students with a stronger sense of interest in a subject are more likely to persist when confronted with frustrating challenges [42, 37; 43], so that strengthening skills in mathematics is a self-feeding cycle. Eccles' [32] discussion of identity development mentions this cycle and state that enjoyable or pleasant experiences with a subject are likely necessary to develop the persistence needed to become an expert in that subject.

3.1.3 Value of Mathematics

Math value is the degree to which a student thinks that math is or will be useful to their life. Like self-concept and interest, value (utility) has been linked to motivation in a number of different research traditions. Among social psychologists, research has shown that value is influenced by self-concept, and, in turn, that value positively influences the kind of goal-setting practices that lead to increased effort [44]. However, research also finds that (perhaps more than self-concept or interest), parents can have a substantial effect on math value [44, 45], which suggests the construct could also be more susceptible to other social pressures or interventions. Cumulatively, these findings suggest that value is often the last component of Math Identity to develop unless external influences (e.g., parents) are involved.

4. Current Study

A number of studies have demonstrated strong links between students' linguistic knowledge and affect (as found in language production), and their success in math. However, to our knowledge, no studies have examined the links between the linguistic features in student language production and variables related to Math Identity. In the current study, we attempt to replicate previous studies that have investigated how linguistic features and affective aspects of students' language production can predict success. More importantly, we also derive models of math identify based on student survey responses related to math value, interest, and self-concept. To derive our language features of interest, we analyzed the language produced by students sending email messages to a virtual pedagogical agent within an online math tutoring system. We analyzed the language using a number of NLP tools in order to extract language information related to text cohesion, lexical sophistication, and sentiment. While our primary interest is in using NLP features to predict variables related to math value, interest, and self-concept, we are also interested in studying the links between NLP features and accuracy scores on beginning level math problems within the online tutoring system. Thus, in this study, we address two research questions:

1. Are linguistic features significant predictors of self-reported student traits related to math value, interest, and self-concept?
2. Are linguistic factors significant predictors of math performance in an on-line tutoring environment?

5. METHOD

5.1 Reasoning Mind

We collected data from Reasoning Mind's *Foundations* product, which is a blended learning mathematics program used in grades 2-5. *Foundations* students learn math in an engaging, animated world at their own pace, while teachers use the system's real-time data to provide one-on-one and small-group interventions [46]. The algorithms and pedagogical logic underlying *Foundations* (previously called *Genie 2*) are described in detail by Khachatryan et al. [11].

The main study mode in *Foundations*, *Guided Study*, consists of a sequenced curriculum divided into objectives, each of which introduces a new topic (e.g., the distributive property) using interactive explanations, presents problems of increasing difficulty on the topic, and reviews previously studied topics. Within *Guided Study*, every student completes problems addressing the basic knowledge and skills required in the objective. These basic problems (known as A Level problems) typically require only a single step to solve and are the lowest of three possible difficulty levels. Students who do well on A Level problems may also proceed to problems of higher difficulty that require two or three steps to solve (B Level and C Level problems) within the objective. They may also access the higher-level problems in an independent study mode called *Wall of Mastery*. Other modes in *Foundations* allow students to play math games against classmates, tackle challenging problems and puzzles, and use points earned by solving math problems to buy virtual prizes.

Foundations uses animated characters to provide a backstory to the mathematics being learned and to deliver emotional support. The main character is the Genie, a pedagogical agent who encourages students throughout their work in the system. Students are also able to send emails to the Genie. These messages are answered in character by part-time Reasoning Mind employees who reference an extensive biography of the Genie and project a consistent, warm, and encouraging persona, model a positive attitude toward learning, and emphasize the importance of practice and challenging work for success. The Genie email system is a popular component of the system, having received 129,879 messages from 38,940 different students in the 2016-17 academic year.

5.2 Participants

The students sampled in this study came from a large sample of *Foundations* students in the 2016-17 academic year, who had written messages for the Genie in the email system. The dates sampled were from August 1, 2016 to June 17, 2017. There were a total of 34,602 such students. The students were from 462 different schools located in 99 different districts, most of which were located in Texas. This analysis samples students in 4th-5th grades because their writing skills are developed enough to be captured by NLP tools. We also included only those students that had completed the post-test survey (discussed in the next sub-section) and those students that had attempted A Level problems. This subset of the data consisted of 970 students.

5.3 Survey Data

The measures used in the present study consisted of three 4-point scales adapted from [47] and administered at the start/end of the 2016/2017 school year. The first was *mathematics self-concept*, which comprised five items that captured the degree to which the student see themselves as a “math person” (e.g., “I have always been good at math”). The second was *interest in mathematics*, which consisted of three items that capture intrinsic curiosity or enjoyment of mathematics (e.g., “How much do you like math?”). The last scale measured *value of mathematics* and consisted of five items that captured the degree to which students find math to be useful (e.g., “How important is it to you to get good grades in math class?”). The Cronbach α of these scales were 0.72, 0.69, and 0.72, respectively.

5.4 Final Corpus

Our language sample for this analysis consisted of messages sent from the students to the Genie. Because many messages contained few words, we aggregated all e-mails sent by each student to create a representation of an individual student’s linguistic activity.

We then implemented data cleaning procedures to reduce the amount of noise in the data. First, all the data was cleaned of non-ASCII characters that could interfere with the NLP tools. Second, all texts were automatically spell-checked and corrected using an open-source Python spelling correction library, in addition to several Python text-cleaning scripts that we developed. Furthermore, several measures were taken to clean the texts, including removing random, non-math symbols such as “#”, “@”, and “&”, as well as omitting repeating words, excessively long words, words with repeating characters, such as “wooorrrddd”, and mixed-type words, such as “$word$”, (with the exceptions of currencies, percentages, timestamps, and ordinals). Next, all non-dictionary, invalid words were removed from the data. This was accomplished by first checking each word against synsets in WordNet, and if a match could not be found, then checking if it consisted of all consonants (always invalid), or if any pair of characters (digraph) in the word were invalid in the English language. Words that met either two of these conditions were removed. Lastly, all texts were cleaned of repeating, non-overlapping groups of words, such as “this word this word this word”. Only word groups of lengths two, three, and four were removed by this approach.

Finally, we removed data from students who had produced fewer than 150 words in writing to the Genie (calculated after cleaning). This cut-off ensures that students produced a large enough language sample to provide a clear representation of their linguistic ability including bag-of-word assumptions for Latent Dirichlet Allocation (LDA) analyses. This left us with data from 351 students for analyses.

5.5 Natural Language Processing Tools

We used several NLP tools to assess the linguistic features in the aggregated posts of sufficient length. These included the Tool for the Automatic Analysis of Lexical Sophistication (TAALES) [48], the Tool for the Automatic Analysis of Cohesion (TAACO) [49], the Tool for the Automatic Analysis of Syntactic Sophistication and Complexity (TAASSC) [50], and the SEntiment ANalysis and Cognition Engine (SEANCE) [51]. In addition, we developed specific indices related to topics commonly discussed with the Genie e-mail system using Latent Dirichlet Allocation (LDA). Thus, the selected NLP features consisted of language variables related to lexical sophistication, text cohesion, syntactic complexity sentiment analysis, and topic similarity respectively. The features are discussed in greater detail below.

5.5.1 TAALES

TAALES reports on a number of indices related to basic lexical information (e.g., the number of tokens, and types), lexical frequency, lexical range, lexical registers, word information features (e.g., concreteness, meaningfulness, polysemy [the number of senses a word has]), and psycholinguistic variables. For instance, the tool uses the Kucera-Francis corpus to compute the number of registers (e.g., humor academic, or fiction registers) that words occur in (a measure of register specificity). The tool also reports on a number of phonological, orthographic, and phonographic neighborhood effects that calculate how many near neighbors based on sound or spelling that a word has. TAALES also reports on variables that measure how long a word takes to name, how accurately words are pronounced, and how many senses a word contains (i.e., polysemy).

5.5.2 *TAACO*

TAACO incorporates a variety of classic and recently developed indices related to text cohesion. For a number of indices, the tool incorporates the Stanford part of speech (POS) tagger [52] and synonym sets from the WordNet lexical database [53]. TAACO provides linguistic counts for both sentence and paragraph markers of cohesion and incorporates WordNet synonym sets. Specifically, TAACO calculates type token ratio (TTR) indices, sentence overlap indices that assess local cohesion, paragraph overlap indices that assess global cohesion, and a variety of connective indices such as logical connectives (e.g., *also, next, so*) and sentence linking connectives (e.g., *but, if, then*).

5.5.3 *TAASSC*

TAASSC measures large and fined grained clausal and phrasal indices of syntactic complexity and usage-based frequency/contingency indices of syntactic sophistication. TAASSC includes indices measured by Lu's [54] Syntactic Complexity Analyzer (SCA) and a number of pre-developed fine-grained indices or clausal complexity and phrasal complexity, The SCA measures are classic measures of syntax based on t-unit analyses [19] where t-units are defined as a dominant and subordinate clause. For instance, SCA measures the number of complex t-units in a text (i.e., T-units that includes both an independent and a dependent clause). The fine-grained clausal indices calculate the average number of particular structures per clause and dependents per clause. The fine-grained phrasal indices measure noun phrase types and phrasal dependent types.

5.5.4 *SEANCE*

SEANCE is a sentiment analysis tool that relies on a number of pre-existing sentiment, social positioning, and cognition dictionaries. SEANCE contains a number of pre-developed word vectors that measure sentiment, cognition, and social order. These vectors are taken from freely available source databases. For many of these vectors, SEANCE also provides a negation feature (i.e., a contextual valence shifter) that ignores positive terms that are negated (e.g., not happy). SEANCE also includes a part of speech (POS) tagger. Examples of affective variables reports by SEANCE include positive and negative polarity metrics, terms related to arousal (as compared to calmness), and respect terms. Cognition examples include words related to socially defined ways of doing work, acts and methods to accomplish goals, time and space, and quantity.

5.5.5 *Latent Dirichlet Allocation (LDA) features*

We developed measures of domain topicality for the messages found in the corpus using LDA. LDA is a computational modeling technique used to infer underlying topics through a generative probabilistic process. We conducted an LDA analysis on the entire corpus of student messages to the Genie and fit 200 topics to the data - the optimal number of topics was inferred using Hierarchical Dirichlet processes [55]. Using these latent topics, each word is perceived as a probability distribution across all topics; if irrelevant for a topic, the corresponding weight is 0, whereas more relevant topics for a given word have higher probabilities. These word weights were then used to create topic distributions for each student in order to identify how strongly student language overlapped with topics covered in the entire Genie message corpus.

5.6 Statistical Analysis

We first calculated correlations between the students' accuracy on A Level problems and their survey scores for Math Identity (self concept, interest, and value). These relationships allow us to better understand how basic math skills interacted with student survey responses for Math Identity.

We followed this up by calculating linear models to assess the degree to which linguistic features in the students' emails to the Genie, along with other behaviors (e.g., question/note posted, questions answered, site visits) were predictive of students' math skills and their self-reported Math Identity. As part of this analysis, we first checked that all variables were normally distributed. For the linguistic variables, we tested only those variables that showed at least a small effect size ($r > .100$) with the response variable. We also controlled for multicollinearity between all the linguistic and non-linguistic variables ($r \geq .700$) such that if two or more variables were highly similar, all but one of the variables (the one with the strongest relationship with the response variable) were removed from the analysis.

We cross-validated our results by dividing data into training and test sets based on a 67/33 split. We used stepwise linear models on the training set to find the best fitting models for each analysis. After model selection, coefficients were checked for suppression and visual inspection of residuals distribution for non-standardized variables was conducted. To obtain a measure of effect sizes, we computed correlations between the fitted and observed values, resulting in an overall R^2 value for the fixed factors in the training set. The model from the training set was used to derive an r and R^2 value for the test data.

6. RESULTS

6.1 Correlations

Pearson correlations were conducted among the response variables to assess links between Math Identity and math scores. The results, reported in Table 1, indicate that all three Math Identity variables were positively and significantly correlated with performance on A level math problems. Medium effects were found for self-concept. Weak effects were found for interest and value. None of the Math Identity variables were strongly associated with one another (i.e., $r < .500$), although correlations with interest approached that threshold for both self-concept (r = .489) and value (r = .491).

Table 1. Correlations between response variables

Variable	Self-concept	Interest	Value
A level score	0.341**	0.205**	0.145*
Self Concept		0.489**	0.309**
Interest			0.491**

Note * $p < .010$, **p < .001

6.2 Linear Model for Self-Concept

A linear model to predict students' self-concept including linguistic, affect, and click-stream variables yielded a significant model, $F(5, 242) = 2.861$, $p < .001$, $r = .356$, $r^2 = .127$ (see Table 2 for details). Two linguistic variables: *Phonographic neighbors, function words* and *word naming accuracy, function words* were significant predictors as were three affective variables: *Methods and goals words, words related to work,* and *polarity verbs*. No click-stream variables were significant predictors. The combination of the five variables accounted for 13% of the variance in the students' self-concept scores. Cross-validating the model on the test set yielded $r = .371$, $r^2 = .138$, demonstrating that the combination of the five variables accounted for 14% of the variance in the student samples comprising the test set.

Table 2. Linguistic model for predicting self-concept scores

Fixed Effect	Coefficient	Std. Error	t
(Intercept)	61.518	21.309	2.887**
Phonographic Neighbors: Function words	-0.284	0.081	-3.512***
Acts and methods terms to accomplish goals	9.441	3.113	3.033**
Words related to work	-6.609	2.342	-2.822**
Polarity verbs	0.247	0.087	2.857**
Word naming accuracy: Function words	-57.807	21.413	-2.700**

Note * *p* < .050, ** *p* < .010, ***p < .001

6.3 Linear Model for Interest

A linear model using linguistic and click-stream variables to predict students' interest yielded a significant model, $F(4, 218) = 4.943$, $p < .001$, $r = .419$, $r^2 = .176$ (see Table 3 for details).. Four affective variables were significant predictors in the model: *Hu Liu negative terms, power words, arousal ratings, and words related to methods and goal*. No click-stream variables were significant predictors. The combination of the four variables accounted for 17% of the variance in the students' interest scores. Using the model from the training set on the samples in the test set yielded $r = .360$, $r^2 = .130$, demonstrating that the combination of the four variables accounted for 13% of the variance in the student samples comprising the test set.

Table 3. Linguistic model for predicting interest scores

Fixed Effect	Coefficient	Std. Error	t
(Intercept)	3.523	0.137	25.708***
Hu Liu negative terms	-0.928	0.201	-4.612***
Power words	-8.440	3.335	-2.531**
Arousal ratings	-9.407	3.336	-2.820**
Acts and methods terms to accomplish goals	8.056	2.951	2.730**

Note * *p* < .050, ** *p* < .010, ***p < .001

6.4 Linear Model for Value

A linear model to predict students' math value using linguistic and click-stream variables yielded a significant model, $F(3, 217) = 7.843$, $p < .001$, $r = .313$, $r^2 = .098$ (see Table 4 for details).. Three variables were significant predictors in the model: polarity verbs component score (verbs related to polarity, aptitude, and pleasantness), time and space terms, and words related to respect. No click-stream variables were significant predictors. The combination of the three affect variables accounted for 10% of the variance in the students' math value scores. Using the model from the training set on the samples in the test set yielded $r = .303$, $r^2 = .091$, demonstrating that the combination of the five variables accounted for 9% of the variance in the student samples comprising the test set.

Table 4. Linguistic model for predicting value scores

Fixed Effect	Coefficient	Std. Error	t
(Intercept)	3.301	0.082	40.254**
Polarity verbs	0.15	0.048	3.107**
Time/space terms	2.932	1.048	2.799**
Respect words	4.776	2.119	2.254*

Note * *p* < .050, ** *p* < .010, ***p < .001

6.5 Linear Model for Math Success

A linear model to predict math success including linguistic and click-stream variables yielded a significant model, $F(5, 217) = 9.130$, $p < .001$, $r = .417$, $r^2 = .174$ (see Table 5 for details).. Five linguistic variables were significant predictors in the model: *Kucera-Francis categories, phonological neighbors distances, complex t-units, polysemy (adverbs), and quantitative terms*. No click-stream variables were significant predictors. The combination of the five variables accounted for 17% of the variance in the students A level math scores. Using the model from the training set on the samples in the test set yielded $r = .378$, $r^2 = .143$, indicating that the combination of the five variables accounted for 14% of the variance in the student samples comprising the test set.

Table 5. Linguistic model for predicting math scores

Fixed Effect	Coefficient	Std. Error	T
(Intercept)	33.544	15.331	3.508***
Kucera-Francis categories	2.721	0.776	2.12*
Phonological neighbor Levenshtein distances	15.225	7.18	-2.701**
Complex T-units	-5.256	1.946	-3.019**
Polysemy (adverbs)	-1.212	0.401	2.348**
Quantitative terms	62.983	26.82	3.508**

Note * *p* < .050, ** *p* < .010, ***p < .001

7. DISCUSSION AND CONCLUSION

Investigating the degree to which students identify with math (e.g., their Math Identity) can provide important information related to student-level differences which in turn could allow for personalization efforts within educational settings. The purpose of this study was to examine links between students' self-reported Math Identity (e.g., math self-concept, value, and interest) and language features found in student e-mails within an on-line math tutoring system. The study also examined links between student math scores and self-reported Math Identity and between math scores and language features. Overall, we find weak to medium relationships between Math Identity variables and math scores. Additionally, language features were able to explain a significant amount of variance for each Math Identity variable and for student math scores. These findings are discussed below along with implications for better understanding Math Identity and developing pedagogical interventions within Reasoning Mind's *Foundation* system.

Our first analysis examined links between A level math scores within the *Foundations* system and student's self-reported Math Identity variables (self concept, interest and value). All of the Math Identity variables were positively correlated with each other as well as with the math-performance metric, although this effect was stronger for self-concept than for interest or value. The correlation matrix in Table 1 provides evidence that the Math Identity variables self-reported by the students were related to math ability within the system.

Our next goal was to investigate if linguistic models could be developed for each of the Math Identity variables. Specifically, we were interested in examining links between the words and language structures produced by the student in their e-mails to the Genie and their self-ratings of self-concept, interest, and value. Our model of student ratings for self-concept explained 14% of the variance in the test set (r = .371). The model was informed by five language features. Three sentiment and cognition features were reported by SEANCE while two features related to lexical sophistication were reported by TAALES. Polarity verbs were again positively related to a math identify variable indicating that students who used more positive verbs reported higher math self-concept. Additionally, students who produced more words related to accomplishing goals (e.g., *build, make,* and *formulate*) reported higher self-concept. Conversely, words related to ways of doing work were negatively associated with self-concept. This may be an effect of the word *grade,* which is included in this category and was common in the e-mails (i.e., students worried about low grades). Two lexical indices for function words were also negatively predictive of self-concept scores: phonographic neighbors and word naming accuracy. These findings suggest that students with higher self-concept produced function words that had fewer neighbors and lower word naming accuracy. In both cases, the results indicate that students producing more sophisticated function words had greater self-concept.

Our model for math interest explained 13% of the variance in the test set (r = .360) and included only sentiment and cognition variables reported by SEANCE. These variables indicate that students with greater math interest used fewer negative terms, fewer words related to arousal (i.e., more words related to calmness), and more words related to acts and methods to accomplish goals, which was also a predictor of self-concept scores. Lastly, words related to power yielded a negative co-efficient with math interest scores. This finding suggests that students that use power words (e.g., *force* and *command*) have lower interest in math.

With respect to students' ratings of their math value, language features were able to predict about 9% of the variance in student test set ratings. (r = .303). Three features were positive predictors of value: polarity verbs, time/space terms, and respect terms. All variables were reported by SEANCE and were related to either sentiment or cognition. The results show that students that reported higher math value produced language in their e-mails that included more positive verbs and showed greater respect through the use of terms such as *honor, admire,* and *respect.* In addition, these students produced more words related to time and space. Time words include prepositions such as *across* and *above* but also space verbs that may be related to math concepts including *circle, curve,* and *distance*.

Finally, we developed a model to predict math success (i.e., scores on A Level problems). This model explained 14% of the variance in math scores (r = .378) using lexical features, a measure of syntactic complexity, and a measure of cognition. The three lexical indices included the number of registers in which a word occurs, phonological neighbors based on Levenshtein distances (i.e., words that words that require more substitutions, insertions, or deletion operations to transform that word into its closest phonologic neighbors), and the polysemy value of adverbs. The first index suggests that students with high math scores produced words that were found across a variety of registers. The second and third indices indicate that students with higher math scores produced more sophisticated language (i.e., adverbs with fewer senses and words that required more operations to find a phonological neighbor). Students with higher math scores also produced fewer complex sentences (sentences with an independent and dependent clause) and used more quantitative words.

Overall, the findings suggest that language variables related to sentiment and cognition can explain a significant amount of the variance in a number of self-reported survey variables related to math self-concept, interest, and value. These variables have the potential to not only better explain the constructs of Math Identity, but also have the potential to be useful for student interventions.

The findings from this study indicate that students who produce more positive language e-mails within the *Foundations* system are more likely to have a positive Math Identity. Conversely, those that use more negative language are more likely to have lower Math Identity. However, it is not just positive and negative terms that are related to Math Identity. Students with stronger Math Identity use more respectful language, less power-related language, and language that is more calm. Lastly, students with stronger Math Identity were more likely to use more sophisticated words or words related to accomplishing goals.

The findings from this study also suggest little overlap between the language features that predict Math Identity and those that predict math success even though we see links between our Math Identity variables and math success within the system. While there are some similarities between self-concept scores and math scores with respect to phonological neighbors, these features differ in their parts of speech (content versus function words). In general, most predictors of math success are related to linguistic features (lexical, syntactic, and cohesion features) while predictors of Math Identity are related to sentiment and cognition features. In total, these sentiment and cognition features provide a profile of students within the system that have high math interest.

Using the models reported here, a number of different interventions could be developed for students identified as likely having low math interest. These interventions could be as simple as having the Genie send an e-mail to students that provides statistics on their successes within the system, their perseverance in answering problems, or simply the number of problems they have attempted or accurately solved over a specific time period. Students could also be asked to correspond with the Genie using metacognitive strategies related to self-assessment and goal-setting activities, as this corresponds with both the interest models we developed here and with long-standing interventions designed to support self-efficacy and interest (cf. [21]). Interventions such as these may assist students in more critically thinking about themselves in relation to math and in better understanding their math knowledge and acquisition.

While the Math Identity profiles developed should be strong enough to drive interventions, the models report only medium effect sizes. Thus, much variance remains to be identified within the existing survey data. Some of that variance may emerge in language features that are not yet captured by NLP tools, while other variance may be related to demographic or other click-stream data available within the system such as the number of messages sent and received by the students within the e-mail system, hours spent on-line within the tutoring system, and number of objectives met within the system. Thus, the findings here should be seen as preliminary with implications for future development.

8. ACKNOWLEDGEMENTS

The authors are indebted to Victor Kostyuk for his help in organizing the data here. In addition, the authors than Stefan Slater for helping with final touches. This research was supported in part by the National Science Foundation (DRL- 1418378). Ideas expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

9. REFERENCES

[1] Corbett, A.T., McLaughlin, M.S., and Scarpinatto, K.C. 2000. Modeling student knowledge: cognitive tutors in high school and college. *User Modeling and User-Adapted Interaction* 10 (Jun. 2000), 81-108. DOI= https://doi.org/10.1023/A:102650562

[2] Romero, C., Ventura, S., Delgado, J. A., and De Bra, P. 2007. Personalized links recommendation based on data mining in adaptive educational hypermedia systems. In *European Conference on Technology Enhanced Learning*, EC-TEL 2007. Springer, Berlin, Heidelberg, 292-306.

[3] Baker, R.. and Ocumpaugh, J. 2014. Interaction-Based Affect Detection in Educational Software. In *The Oxford Handbook of Affective Computing*, R. Calvo, S. D'Mello, J. Gratch, A. Kappas, Eds. Oxford U. Press, Oxford, UK.

[4] 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, 1-2, 81-123. DOI= https://doi.org/10.1007/s11257-007-9040-y

[5] Cooper, D. G., Arroyo, I., Woolf, B. P., Muldner, K., Burleson, W., and Christopherson, R. 2009. Sensors model student self concept in the classroom. In *International Conference on User Modeling, Adaptation, and Personalization*, 30-41.

[6] Winne, P. H. and Baker, R. S. 2013. The potentials of educational data mining for researching metacognition, motivation and self-regulated learning. *Journal of Educational Data Mining* 51 (May 2013), 1-8.

[7] Pennebaker, J. W. and King, L. A. 1999. Linguistic styles: Language use as an individual difference. *Journal of Personality and Social Psychology* 77, 6 (Dec. 1999), 1296-1312.

[8] Pennebaker, J. W. and Graybeal, A. 2001. Patterns of natural language use: Disclosure, personality, and social integration. *Current Directions in Psychological Science* 103 (Jun. 2001), 90-93. DOI= https://doi.org/10.1111/1467-8721.00123

[9] Schlenker, B. and Weigold, M. 1989. Goals and the self-identification process: Constructing desired identities. In *Goal Concepts in Personality and Social Psychology*, Pervin, Ed.. Lawrence Erlbaum Assoc., Hillside, NJ, 243-89.

[10] Nosek, B., Banaji, M., and Greenwald, A. 2002. Math= male, me= female, therefore math≠ me. *Journal of Personality and Social Psycholog*y 83, 1 (Jul. 2002), 44-59.

[11] Khachatryan, G., Romashov, A., Khachatryan, A., Gaudino, S., Khachatryan, J., Guarian, K., and Yufa, N. 2014. Reasoning Mind Genie 2: An intelligent tutoring system as a vehicle for international transfer of instructional methods in mathematics. *International J. A.I. Ed*. 243, 3 (Sep. 2014), 333-382. DOI= https://doi.org/10.1007/s40593-014-0019-7

[12] Ryan, K. and Ryan, A. 2005. Psychological processes underlying stereotype threat and standardized math test performance. *Edu'nal Psychologist* 40, 1 (Jun. 2010), 53-63. DOI= https://doi.org/10.1207/s15326985ep4001_4

[13] MacGregor, M. and Price, E. 1999. An exploration of aspects of language proficiency and algebra learning. *Journal for Research in Math Education* 30, 4 (Jul. 1999), 449–467. doi: 10.2307/749709

[14] Vukovic, R. K. and Lesaux, N.K. 2013. The relationship between linguistic skills and arithmetic knowledge. *Learning and Individual Differences* 23 (Feb. 2013), 87-91. DOI= https://doi.org/10.1016/j.lindif.2012.10.007

[15] Hernandez, F. 2013. The Relationship Between Reading and Math Achievement of Middle School Students as Measured by the Texas Assessment of Knowledge and Skills. Doctoral Thesis.

[16] LeFevre J., Fast, L., Skwarchuk, S., Smith-Chant, B., Bisanz, J., Kamawar, D., and Penner-Wilger, M. 2010. Pathways to math: Longitudinal predictors of performance. *Child Development* 81, 6 (Nov. 2010), 1753–1767. DOI= 10.1111/j.1467-8624.2010.01508.x

[17] Crossley, S. A., Liu, R., and McNamara, D. S. 2017. Predicting math performance using natural language processing tools. *Proceedings of the 7th International Learning Analytics and Knowledge LAK Conference*. LAK'17. ACM, New York, NY, 339-347.

[18] Crossley, S.A., Barnes, T., Lynch, C., and McNamara, D.S. 2017. Linking language to math success in a blended course. In *Proceedings of the 10th International Conference on Educational Data Mining* (Wuhan, China), Hu, X., Barnes, T., Hershkovitz, A., and Paquette, L. Eds. 180-185

[19] Crossley, S. A. and Kostyuk, V. 2017. Letting the Genie out of the Lamp: Using Natural Language Processing tools to predict math performance. In *Language, Data, and Knowledge LDK 2017*, Gracia J., Bond F., McCrae J., Buitelaar P., Chiarcos C., and Hellmann S. Eds. In. *Lecture Notes in Computer Science*, vol 10318. Springer, Cham, Switzerland.

[20] Syed, M., Azmitia, M., and Cooper, C. 2011. Identity and academic success among underrepresented ethnic minorities: An interdisciplinary review and integration. *Journal of Social Issues* 67, 3 (Sep. 2011), 442-468. DOI= 10.1111/j.1540-4560.2011.01709.x

[21] Bandura, A. 1977. Self-efficacy: toward a unifying theory of behavioral change. *Psychological Review* 84, 2, 191-215. DOI= http://dx.doi.org/10.1037/0033-295X.84.2.191

[22] Bem, S. 1974. The measurement of psychological androgyny. *J. of Consulting and Clinical psychology* 42, 2, 155-162. DOI= http://dx.doi.org/10.1037/h0036215

[23] Hitlin, S. 2003. Values as the core of personal identity: Drawing links between two theories of self. *Social Psychology Quarterly* 66, 2 (Jun. 2003), 118-137. DOI= 10.2307/1519843

[24] Erikson, E. 1968. *Youth: Identity and Crisis*. Norton & Company, *New York, NY*.

[25] Syed, M., Azmitia, M., & Cooper, C. R. 2011. Identity and academic success among underrepresented ethnic minorities: An interdisciplinary review and integration. *Journal of Social Issues, 67*(3), 442-468.

[26] San Pedro, M.O.Z., Baker, R.S.J.d., Bowers, A., Heffernan, N. 2013. Predicting College Enrollment from Student Interaction with an Intelligent Tutoring System in Middle School. In *Proc. 6th International Conf. on Educational Data Mining*, 177-184.

[27] San Pedro, M.O.Z., Ocumpaugh, J., Baker, R., Heffernan, N. 2014. Predicting STEM and Non-STEM College Major Enrollment from Middle School Interaction with Mathematics Educational Software. In *Proc. 7th International Conf. on Educational Data Mining*, 276-279.

[28] Landers, M. 2013. Buying in and checking out: Identity development and meaning making in the practice of mathematics homework. *Qualitative Research in Ed.* 2*,* 2, 130-160.

[29] Cadinu, M. and Galdi, S. 2012. Gender differences in implicit gender self-categorization lead to stronger gender self-stereotyping by women than by men. *European Journal of Social Psychology* 4, 25 (Apr. 2012), 546-551. DOI= 10.1002/ejsp.1881

[30] Keller, J. 2007. Stereotype threat in classroom settings: The interactive effect of domain identification, task difficulty and stereotype threat on female students' maths performance. *British J. of Ed. Psych.* 77, 2 (Jun/ 2017), 323-338. DOI= 10.1348/000709906X113662

[31] Steele, C. 1997. A threat in the air. How stereotypes shape intellectual identity and performance. *Am. Psychologist* 52, 6, 613-629.

[32] Eccles, J. 2009. Who am I and what am I going to do with my life? Personal and collective identities as motivators of action. *Educational Psychologist* 44, 2 (Apr. 2009), 78-89. DOI= https://doi.org/10.1080/00461520902832368

[33] Bong, M. and Skaalvik, E. M. 2003. Academic self-concept and self-efficacy: How different are they really?. *Educational Psychology Review* 15, 1 (Mar. 2003), 1-40. DOI= https://doi.org/10.1023/A:1021302408382

[34] Pajares, F. and Miller, M. D. 1994. Role of self-efficacy and self-concept beliefs in mathematical problem solving: A path analysis. *Journal Of Educational Psychology* 86, 2, 193-203.

[35] Epstein, S. 1973. The self-concept revisited: Or a theory of a theory. *American Psychologist* 28, 5, 404-416. DOI= http://dx.doi.org/10.1037/h0034679

[36] Shavelson, R. and Bolus, R. 1982. Self concept: The interplay of theory and methods. *J. Educational Psychology* 74, 1, 3-17. DOI= http://dx.doi.org/10.1037/0022-0663.74.1.3

[37] Hidi, S. and Renninger, K. 2006. The four-phase model of interest development. *Ed. Psychologist* 41, 2 (Jun. 2010), 111-127. DOI= https://doi.org/10.1207/s15326985ep4102_4

[38] Bandura, A. and Schunk, D. 1981. Cultivating competence, self-efficacy, and intrinsic interest through proximal self-motivation. *J. of Personality and Social Psych*. 41, 3, 586-598. DOI= http://dx.doi.org/10.1037/0022-3514.41.3.586

[39] Sansone, C., Weir, C., Harpster, L., and Morgan, C. 1992. Once a boring task always a boring task? Interest as a self-regulatory mechanism. *J. of Personality and Social Psych.* 63, 3 (Sep. 1992), 379-390.

[40] Roberts, B. and DelVecchio, W. 2000. The rank-order consistency of personality from childhood to old age: A quantitative rev. of longitudinal studies. *Psych. Bulletin* 126, 1 (Jan. 2000), 3-25.

[41] Campbell, N. and Hackett, G. 1986. The effects of mathematics task performance on math self-efficacy and task interest. *J. of Vocational Behavior* 28, 2, 149-162. DOI= https://doi.org/10.1016/0001-8791(86)90048-5

[42] Fink, R. P. 1998. Interest, gender, and literacy development in successful dyslexics. In L. Hoffmann, A. Krapp, K. A. Renninger, & J. Baumert (Eds.), *Interest and learning: Proceedings of the Seeon Conference on interest and gender* (pp. 402–407). Kiel, Germany: IPN.

[43] Prenzel, M. 1992. The selective persistence of interest. In *The Role of Interest in Learning and Development*, Renninger, K. A.. S. Hidi, and A. Krapp, Eds. Lawrence Erlbaum, Hillsdale, NJ, 71–98.

[44] Chouinard, R., Karsenti, T., and Roy, N. 2007. Relations among competence beliefs, utility value, achievement goals, and effort in mathematics. *British Journal of Educational Psychology* 77, 3 (Sep. 2007), 501-517. DOI= 10.1348/000709906X133589

[45] Harackiewicz, J., Rozek, C., Hulleman, C., and Hyde, J. 2012. Helping parents to motivate adolescents in mathematics and science: An experimental test of a utility-value intervention. *Psychological Science* 23, 8 (Jul. 2012), 899-906. DOI= https://doi.org/10.1177/0956797611435530

[46] Miller, W., Baker, R., Labrum, M., Petsche, K., Liu, Y-H., and Wagner, A. 2015. Automated Detection of Proactive Remediation by Teachers in Reasoning Mind Classrooms. In *Proc. 5th International Learning Analytics and Knowledge Conf*. (Poughkeepsie, NY, March 16 - 20, 2015). AMC, New York, NY, 290-294.

[47] Mingle, L. (2013). *Threats to success in mathematics: examining the combined effects of choking under pressure and stereotype threat* (Doctoral dissertation, University of Illinois at Urbana-Champaign).

[48] Kyle, K., Crossley, S. A., and Berger, C. 2017. The tool for the automatic analysis of lexical sophistication (TAALES): version 2.0. *Behavior research methods,* 1-17. DOI= https://doi.org/10.3758/s13428-017-0924-4

[49] Crossley, S.A., Kyle, K. and McNamara D.S. 2016. The tool for the automatic analysis of text cohesion (TAACO): Automatic assessment of local, global, and text cohesion. *Behavior Research Methods* 28, 4 (Sep. 2015), 1227-1237. DOI= https://doi.org/10.3758/s13428-015-0651-7.

[50] Kyle, K. and Crossley, S. A. 2017. Assessing syntactic sophistication in L2 writing: A usage-based approach. *Language Testing* 34, 4 (Sep. 2017), 513–535. DOI= https://doi.org/10.1177/0265532217712554

[51] .Crossley, S.A., Kyle, K. and McNamara D.S. 2016. Sentiment Analysis and Social Cognition Engine (SEANCE):

An automatic tool for sentiment, social cognition, and social order analysis. *Behavior Research Methods* 49, 3 (Jun. 2017), 803-821. DOI= https://doi.org/10.3758/s13428-016-0743-z.

[52] Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. ACL, Baltimore, MA, 55–60

[53] Miller, G. A. 1995. WordNet: A lexical database for English. *Communications of the ACM* 38, 11 (Nov. 1995), 39–41. DOI= 10.1145/219717.219748

[54] Lu, X. 2010. Automatic analysis of syntactic complexity in second language writing. *International Journal of Corpus Linguistics* 15, 4, 474-496.

[55] Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. 2006. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association* 101, 1566–1581.

Modeling Hint-Taking Behavior and Knowledge State of Students with Multi-Task Learning

Ritwick Chaudhry[*†]
Indian Institute of Technology
Bombay
Mumbai, India
ritwickchaudhry@gmail.com

Harvineet Singh[*]
Adobe Research
Bengaluru, India
harvines@adobe.com

Pradeep Dogga[†]
Indian Institute of Technology
Kharagpur
Kharagpur, India
pradeepdogga@gmail.com

Shiv Kumar Saini
Adobe Research
Bengaluru, India
shsaini@adobe.com

ABSTRACT

Interactive learning environments facilitate learning by providing hints to fill the gaps in the understanding of a concept. Studies suggest that hints are not used optimally by learners. Either they are used unnecessarily or not used at all. It has been shown that learning outcomes can be improved by providing hints when needed. An effective hint-taking prediction model can be used by a learning environment to make adaptive decisions on whether to withhold or provide hints. Past work on student behavior modeling has focused extensively on the task of modeling a learner's state of knowledge over time, referred to as knowledge tracing. The other aspects of a learner's behavior such as tendency to use hints has garnered limited attention. Past knowledge tracing models either ignore the questions where a hint was taken or label hints taken as an incorrect response. We propose a multi-task memory-augmented deep learning model to jointly predict the hint-taking and the knowledge tracing task. The model incorporates the effect of past responses as well as hints taken on both the tasks. We apply the model on two datasets – ASSISTments 2009-10 skill builder dataset and Junyi Academy Math Practicing Log. The results show that deep learning models efficiently leverage the sequential information present in a learner's responses. The proposed model significantly out-performs the past work on hint prediction by at least 12% points. Moreover, we demonstrate that jointly modeling the two tasks improves performance consistently across the tasks and the datasets, albeit by a small amount.

1. INTRODUCTION

E-learning is changing knowledge creation and sharing in a profound way by bringing personalized learning experiences to a learner's device. Assessments in the form of quizzes or assignments form an important component of an e-learning software. A personalized e-learning environment identifies the gaps in understanding of a concept and effectively uses learning aids such as hints to fill these gaps. *Knowledge tracing* is the task of estimating a learner's state of knowledge over time with the goal of predicting the performance of the learner in future assessments. Knowledge tracing is used for deciding which question to ask in an adaptive learning environment. Current set of knowledge tracing models neither incorporate the effect of a learning aid on the level of understanding of a concept nor predict whether a learner is likely to use a learning aid.

A learning aid, common to many interactive learning environments, is the option to take a hint during an assessment [3]. However, the data shows that learners tend to use hints inappropriately. One problem is that of abusing hints [2]. They tend to spend less time on solving the assessment and opt for hint without attempting to solve the problem. Figure 1 shows the percentage of responses with correct answers, incorrect answers, and percent directly opted for hint by each question. The x-axis is sorted by the percent of correct responses for a question in increasing order. The data for this chart is from ASSISTments dataset [14] for 2009-2010.[1] As expected, % hint taken is negatively correlated with % correct. In other words, more learners tend to take hints on difficult questions. However, as Figure 2 shows, the hint takers tend to spend less time on a question than the learners who attempt the question, irrespective of whether the question is correctly or incorrectly answered. The research on this subject shows that the learners who attempt a question tend to have a higher probability of achieving proficiency in the subject [19]. Also, the learners who use hints very frequently tend to have the lowest learning rate [13]. Section 3 presents a review of the literature on hints as a learning aid. The literature shows that hints are an important learning aid but offering hints indiscriminately can lead to poor learning outcomes. A personalized e-learning

[*] These authors contributed equally

[†] Work done during an internship at Adobe Research

[1] The dataset is available at https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data/skill-builder-data-2009-2010.

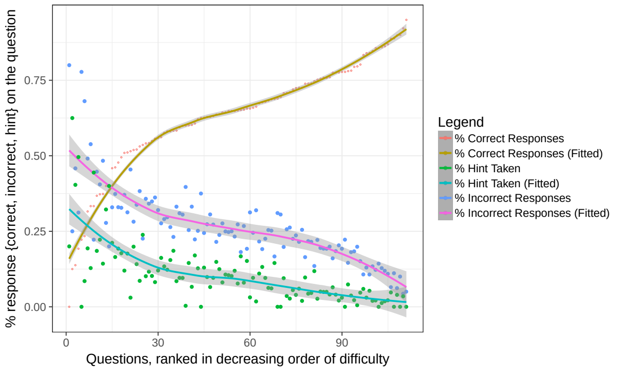


Figure 1: Percent of correct attempts, incorrect attempts, hints opted for each question in ASSISTments data. The questions are sorted by % correct responses.

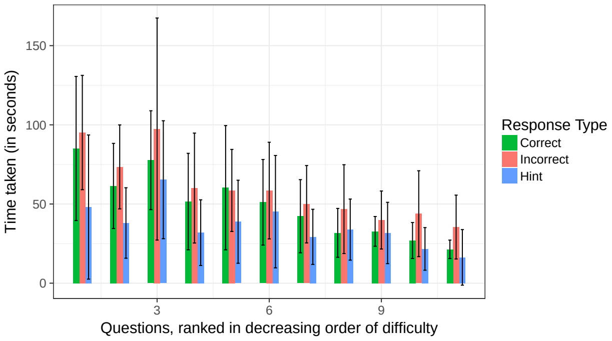


Figure 2: The box plot shows the distribution of time taken to attempt a question when response was correct (in green color), incorrect (in red), and when hint was taken (in blue). x-axis is sorted from lowest % correct on left to highest in right.

environment can use likelihood of taking a hint and the effect of taking a hint on learning to decide whether to show a hint. For example, the environment can proactively suggest hints to students who are stuck with a concept and have a low likelihood of taking a hint themselves.

Another reason to model the hint-taking behavior is to improve the performance of a knowledge tracing model. The existing knowledge tracing models do not model the hint-taking behavior. Section 2 presents the past work on knowledge tracing and hint-taking prediction. Traditional knowledge tracing models either tag a hint taken as an incorrect response or remove the data point where hints were taken. The two responses, i.e. attempting to solve a question and taking a hint directly, tend to result in different learning outcomes. Hence, conflating an incorrect response with a hint taken can deteriorate model performance. We show that explicitly modeling the hint-taking behavior improves performance of the model. Additionally, a higher propensity to take hints might be informative about the likelihood of answering questions correctly [19, 13]. Hence, throwing away the data points where hints were taken is akin to throwing away useful information. Conversely, knowledge tracing tasks contain information about whether a student is likely to take a hint. The synergies between the knowledge tracing and the hint-taking task motivates the application of a multi-task learning model [8]. Another important modeling consideration is the parameterization of the skill level. A knowledge tracing model is parameterized by deciding the level of heterogeneity in a learner's skill level and the question difficulty parameters. In the traditional knowledge tracing models, one might represent the skill level using one common parameter for all concepts or use a different parameter for each concept or a group of concepts clustered based on domain knowledge. Recently, deep learning based models have been used for knowledge tracing [23, 16, 34] which automatically capture the dependencies between different concepts based on the student response sequences. We extend the memory-augmented deep learning model proposed by Zhang *et al.* [34] to include hints taken in the past as an input and the prediction of hint-taking as an auxiliary task. We call this model `Colearn`. Section 4 describes the proposed model. Section 6 describes the evaluation methodology and estimation approach, including how the model hyperparameters are set.

The proposed model is compared with the baseline models from traditional approaches as well as deep learning based approaches. Section 7 describes the baseline models. We perform experiments on two popular datasets – ASSISTments 2009-2010 skill builder dataset and Junyi Academy Math Practicing Log. Section 5 describes the two datasets. Both the datasets contain information on whether a hint was taken. ASSISTments dataset contains the information whether a learner first attempted a question or directly took a hint. However, Junyi dataset contains noisy information on hints taken as it contains information on whether a hint was taken regardless of whether a hint was taken first or the question was attempted prior to it. The importance of this distinction is supported by past studies.

Results show that a memory-augmented deep learning model improves hint prediction performance from 79.10% to 91.12% on ASSISTments dataset and from 77.62% to 92.31%. `Colearn`, which is a multi-task memory-augmented deep learning model, further improves, by a small margin, the performance of the hint-taking prediction task by 0.63% and 0.03% point, respectively for the two datasets. Additionally, `Colearn` improves the performance on the knowledge tracing task for ASSISTment dataset by 0.25% point and for Junyi dataset by 0.18% points. Note that the baseline model for knowledge tracing is another memory-augmented deep learning model. Although the effect on performance is small, a benefit of the joint modeling of the two tasks is that we can work with only one model instead of two while training and scoring.

One of the criticisms of the deep learning based approaches is that the estimated parameters do not enhance our understanding of how the world works. We try to understand the meaning of the estimated parameters, especially the question embedding vectors, in Section 7.3. The analysis shows that a question embedding tends to capture question's difficulty.

In summary, the main contributions of this work are fourfold. First, we show a large improvement in the performance of the hint-taking prediction task by using a memory-augmented deep learning model. Second, we motivate joint modeling of knowledge tracing and hint-taking prediction tasks which have been modeled separately in the prior work. Third, we extend a recent memory-augmented deep learning model for knowledge tracing to the task of hint-taking prediction. The proposed model, `Colearn`, incorporates the sequence of correct, incorrect response as well as hint-taking behavior on past questions as inputs. The model adds the hint-taking prediction as an auxiliary task. Fourth, we extensively evaluate the proposed model on two real-world datasets and show that our approach outperforms the competitive baselines on both the tasks.

2. RELATED WORK

This paper builds on the literature on knowledge tracing and on learning aids such as hints. Knowledge tracing in an interactive learning environment is an extensively studied area. Different approaches have been proposed in past.

Item Response Theory or IRT models the probability that a student answers a question correctly as a function of the following two parameters: one representing the student's skill level and the second representing the question difficulty [12]. The probability that a student answers a question correctly decreases with the question difficulty and increases with the student skill level, all else being equal. The student skill level and question difficulty are scalars which are estimated from data. Recent extensions to IRT, such as Hierarchical IRT, partition questions into groups, e.g. based on concepts covered, and model student skill level and item difficulty for each group separately [30]. However, these models do not use the information present in the sequence of responses. This results in incorrect responses followed by correct responses to be treated the same as the reverse sequence. Intuitively, a knowledge tracing model should put more weight on the performance on recent responses.

Bayesian Knowledge Tracing or BKT is another widely-used model. It uses information in the sequence of responses. BKT uses a Hidden Markov Model with the student skill as the latent variable and the responses as the observed variables [11]. One reason for the popularity of BKT is that, unlike IRT, it models student's skill in each concept separately. This information can be used by a learning system to personalize a learning activity. For example, a learning system can repeat a concept, switch to a new concept or skip a concept altogether based on the estimates of the skill level attained in the concepts.

Deep Learning based approaches have been employed due to the flexibility these approaches provide in modeling the skill of a student and the difficulty level of a question. Piech *et al.* [23] use Long Short-Term Memory (LSTM) cells to model sequence of student responses. They show significant improvement over BKT in predicting the student responses on many datasets. There has been concern voiced due to the lack of interpretability of the Deep Learning based approaches. Khajah *et al.* [16] show that DKT's performance can be matched by modifying BKT model. However, matching DKT's performance required significant domain knowledge on the processes involved in the learning process and insights from DKT model [16]. On the other hand, a Deep Learning based model performs well even without explicitly building a domain specific knowledge into the model. Memory-augmented neural networks, proposed for this task by Zhang *et al.* [34], provide even more flexibility to model student skill and question difficulty. A similar network architecture has been used for question-answering on free-form text documents [20].

Hints as a study help strategy has been extensively studied. The literature on how to provide hints has focused on whether to provide hints on-demand or proactively. Duong *et al.* [13] propose a model incorporating hint usage information in knowledge tracing. However, they do not use this information to predict the probability that a user will take a hint or not. Castro *et al.* [9] use a technique called *tabling method* to predict whether a student will attempt or take a hint in the next question. The model does not consider the complete sequence of student responses in the past and it is difficult to train for the longer sequences. This results in poor performance of the model.

In summary, there is rich literature on predicting the likelihood of a correct response and some recent work in predicting hint usage. However, the literature, to the best of our knowledge, has not modeled these two related problems jointly. Past work on *multi-task learning* (MTL) [8] suggests that adding an auxiliary task can help in improving the performance on both the tasks. MTL has shown considerable benefits in many domains including computer vision [21], natural language processing [17], health diagnostics [35], among others. Our proposed model includes effect of hints on future probability of answering a question correctly. This information can be used to decide when to provide a hint on a particular question.

Our Contribution: We extend the model proposed by Zhang *et al.* [34]. We include the hint usage information by changing the encoding of the inputs to the network. In addition, we add the components which share the network weights for the auxiliary task of predicting the probability of taking a hint. This results in increased prediction accuracy for the tasks of whether the learner will take a hint as well as whether a learner will answer a question accurately.

3. BACKGROUND

There is a large literature on hints as a learning aid that provides motivation for the joint modeling of item response and hint usage. The literature shows that hints are important but prone to misuse if provided indiscriminately. The research also shows that attempting a question and taking a hint directly have different implications for learning a concept.

Mathews *et al.* [19] shows that learners who first attempt to solve a question tend to learn by themselves and have higher probability to master the knowledge. This result has a basis in the theory that the process of attempting a question activates self-explanation, which is an important meta-cognitive skill [4, 10, 7, 22, 25, 29]. While hints are useful learning aid, the research on how hints are used show that easy access to hints may lead to sub-optimal outcomes. In studies of help-seeking from human tutors, it has been found that those who need help the most are the least likely to ask for it [15, 24, 26]. Computer-based help systems can potentially improve the use of help [32]. Given that many learning environments provide some form of on-demand help, it might seem that effective use of help would be an important factor influencing the learning results obtained with these systems. However, there is evidence that learners are not using the help facilities offered by learning environments effectively [3]. They often ignore the help facilities or use them in ways that are not likely to help learning. They frequently use the system's on-demand hints to get answers, without trying to understand how the answers are derived or the reasons behind the answers [1]. It is shown that the learners who opt for hints very frequently tend to have the lowest learning rate [13]. On the other hand, there is also evidence that, when used appropriately, on-demand help in an interactive learning environment can have a positive impact on performance [1, 5] and learning [27, 31, 32]. Also, providing tutoring with respect to student's help-seeking behavior helps them to become better help seekers and thus better future learners [6]. A request for help is appropriate when a student is stuck while solving a tutor problem but not when she has not yet thought about the problem. Further, students should carefully read and interpret the help given by the system. Aleven *et al.* [2] described a model of help-seeking behavior within a cognitive tutor. The authors have created a taxonomy of errors in student's help-seeking behavior. Based on the frequency of the meta-cognitive bugs defined by their model, it was observed that 36% of the actions taken by students were classified as help abuse bugs and 19% of the actions as help avoidance. To make a better tutoring system which can guide the students in regulating their help-seeking behavior, it is essential to incorporate the effect of hints in knowledge tracing. Traditional knowledge tracing models do not take the hint usage into account.

3.1 Notations

Next, we introduce notations for the joint model. Let the interactions of a learner till time T are denoted by $X = (x_1, x_2, x_3, \ldots, x_T)$. Here, each interaction x_t is an encoding representing the tuple (q_t, r_t, h_t) containing an identifier for the question attempted q_t, a binary indicator r_t, encoding the response, and another binary indicator h_t, encoding hint usage. The hint usage variable is positive only if the hint was taken directly instead of attempting the question first. Let $Q = \{q_t\}_t$ be the set of distinct questions. The interaction tuple can contain additional information collected such as time taken to attempt, type of question, concepts involved in the question and so on. The task of a knowledge tracing model is to predict the probability of correctly answering a question $q_{t'} \in Q, t' > T$, i.e. $\text{Prob}(r_{t'} = 1|q_{t'}, X)$. And, the task of predicting a hint usage model is to estimate $\text{Prob}(h_{t'} = 1|q_{t'}, X)$. Both of these tasks are supervised learning problems and can be modeled using a binary classifier. Instead of building two separate models for these tasks, we model them jointly within a deep learning based classification framework.

4. MODEL

Zhang *et al.* [34] proposed a memory-augmented neural network model, called Dynamic Key-Value Memory Networks or `DKVMN`, for knowledge tracing. This model performed better than the baseline models on three real-world datasets. This model is used as a baseline for the proposed multi-task model due its many favorable properties. It does not require extensive feature engineering or metadata information such as mapping of items to skills and the model offers flexibility in adding more tasks as well as inputs. We first give a brief description of their model, followed by our modifications. Reader is referred to Zhang *et al.* [34] for further implementation details regarding the original model.

4.1 Dynamic Key-Value Memory Networks, `DKVMN`

The neural network is designed to store the knowledge state of a learner based on past interactions. This is done using a memory component which works like a key-value store. Each attempted question is mapped to a set of concepts which are the keys in the memory component. The corresponding values are a learner's knowledge state in each of these concepts. The network has a mechanism to update the states because of learner's response to the question. The key-value pairs are modeled using vectors instead of scalars for more representational flexibility. So, for each question the output from the memory component gives a learner's knowledge state. This is compared with the difficulty level of the question, which is the output of another component, to arrive at probability of correctly answering the question. All operations are implemented using differentiable operators like multiplication, addition, sigmoid function on matrices so that the network can be trained end-to-end using gradient descent optimization techniques.

4.2 Proposed Model, `Colearn`

The `DKVMN` model does not consider the effect of taking hints during assessment. It considers hint usage as an incorrect attempt by the learner, as is the standard approach in existing models. However, the update in knowledge state of a learner is different when a question is attempted as opposed to when a hint is taken without any attempt. We modify `DKVMN` to incorporate hint information by changing the input and output layers of the model. Figure 3 shows the modified network. Next, we describe the components of `DKVMN` and our modifications to it.

4.2.1 Input Layer

In the update phase of the model, instead of using one-hot encoding of (q_t, r_t), we encode (q_t, r_t, h_t) into a vector of length $2|Q| + 1$, where Q is the set of distinct questions. The first $|Q|$ dimensions are a one-hot vector representing the correct attempt on the question, i.e. in case of a correct attempt, the vector has 1 at the index of the question and has 0 everywhere else. Similarly, the next $|Q|$ dimensions encode incorrect attempt. The last dimension of the vector is a binary value indicating whether a hint is taken. This input encoding changes the way the value vectors in the memory component are changed due to the information whether a hint is used or not is also present. An example of the input encoding is illustrated in Table 1 where there is a total of two exercises.

We tried different ways of representing the three outcomes, *viz.* correct response, incorrect response, and hint taken. These included one-hot encoding with all three outcomes

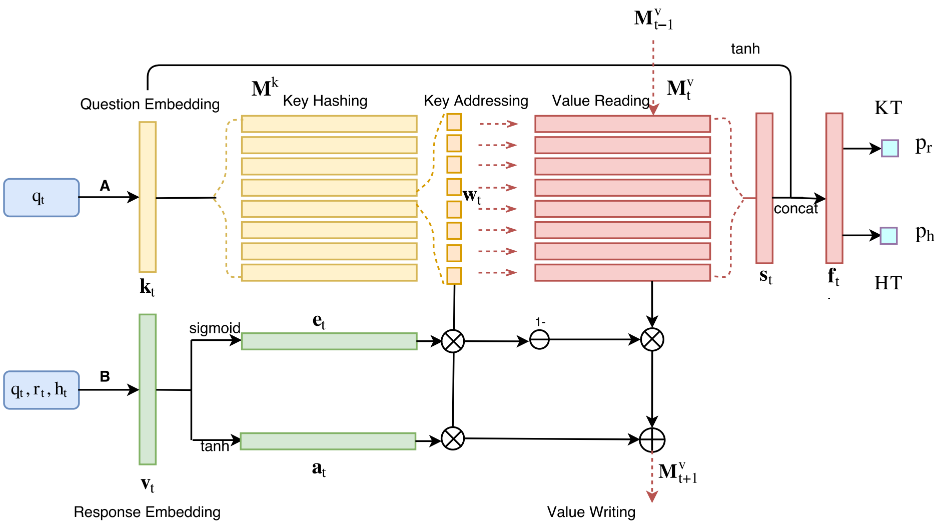


Figure 3: Architecture of the neural network for joint modelling of knowledge state and hint use. KT and HT refer to Knowledge tracing and Hint-Taking tasks.

Response	Encoding	
	`DKVMN`	`Colearn`
Q2-Correct	$(0,1,0,0)$	$(0,1,0,0,0)$
Q2-Incorrect	$(0,0,0,1)$	$(0,0,0,1,0)$
Q2-Hint	–	$(0,0,0,0,1)$
Q1-Hint	–	$(0,0,0,0,1)$

Table 1: Response encoding in case of two exercise tags

with a length of $3|Q|$. The chosen encoding gave the best results in the experiments. This encoding represents response on two different questions where hints are taken with the same vector (see example in Table 1). Since the network already incorporates index of the current question as a separate input, using $|Q|$ extra dimensions for hint encoding in update phase adds more parameters which are not required.

4.3 Key-Value Store

Key-value memory networks, introduced in [20], have an explicit memory component which is an array of pairs of memory slots where each slot is a real-valued vector. Given a query, the relevant information is fetched from the slots using an attention-based mechanism depending on which slots are relevant for that query. The mechanism has three major components which are described next.

- **Key Hashing:** The *key* part of the pairs holds the static information representing the various hidden concepts using vectors. Each of the key vectors $(\mathbf{M}^k(1), \ldots, \mathbf{M}^k(n))$ represents a concept.

- **Key Addressing:** Given the t^{th} question answered by a student, the relevance of each concept in that question is found out using an attention mechanism. Each question is first converted into an embedding

$$\mathbf{k}_t = \mathbf{A}\mathbf{q}_t \tag{1}$$

and the weight of each concept c_i in q_t is given by

$$w_t(i) = \text{Softmax}({\mathbf{k}_t}^T \mathbf{M}^k(i)) \tag{2}$$

where $\mathbf{A}$ is the question embedding matrix, $\mathbf{q}_t$ denotes the one-hot encoded question, $\mathbf{M}^k(i)$ denotes the key vector of the i^{th} concept and $\text{Softmax}(x_i) = e^{x_i} / \sum_j e^{x_j}$. The question embedding vector $\mathbf{k}_t$ obtained from matrix $\mathbf{A}$, the key matrix $\mathbf{M}^k$ are shown in yellow color and attention weight vector $\mathbf{w}_t = (w_t(1), \ldots, w_t(n))$ is shown in orange in Figure 3.

- **Value Reading:** Given the weight $w_t(i)$ of each concept c_i in question q_t given by Equation 2, the student's skill in that question is calculated as the weighted sum of the knowledge in each of the concepts, as taken from value matrix $\mathbf{M}^v_t$. The value matrix is shown in pink color in Figure 3. The student's skill in the question q_t is returned as

$$\mathbf{s}_t = \sum_{i=1}^{n} \mathbf{M}^v_t(i) * w_t(i) \tag{3}$$

This skill is then used to make predictions about the student's response correctness and hint usage.

- **Value Writing:** Once we get student's actual response to the question, knowledge state is updated. This part is shown in green color in Figure 3. The update in each of the concept c_i's value vectors are also weighted according to the calculated weight $w_t(i)$ of the concept (2). The student's response is encoded in a vector, $\mathbf{x}_t$ of size $2|Q|+1$ to represent a correct attempt or an incorrect attempt or a hint taken.

$$\mathbf{x}_t = \text{encoded tuple}(q_t, r_t, h_t)$$

This encoding, described in 4.2, is then converted into an embedding $\mathbf{v}_t$, given by

$$\mathbf{v}_t = \mathbf{B}\mathbf{x}_t$$

where $\mathbf{B}$ is the response embedding matrix. When updating the student's knowledge state, the memory is erased

first before new information is added.
The erase vector $\mathbf{e}_t$ is calculated as

$$\mathbf{e}_t = \text{Sigmoid}(\mathbf{E}^T\mathbf{v}_t + \mathbf{b}_e)$$

where $\mathbf{E}$ is a linear transformation matrix, $\mathbf{b}_e$ is the bias and $\text{Sigmoid}(x_i) = 1/(1 + e^{x_i})$.
The addition vector $\mathbf{a}_t$ is calculated as

$$\mathbf{a}_t = \text{Tanh}(\mathbf{D}^T\mathbf{v}_t + \mathbf{b}_a)$$

where $\mathbf{D}$ is a linear transformation matrix, $\mathbf{b}_a$ is the bias and $\text{Tanh}(x_i) = (e^{x_i} - e^{-x_i})/(e^{x_i} + e^{-x_i})$.
After the t^{th} response, the value matrix is updated as

$$\mathbf{M}^v_t(i) = \mathbf{M}^v_{t-1}(i) \odot [1 - w_t(i)\mathbf{e}_t] + w_t(i)\mathbf{a}_t$$

Thus, the model adds and forgets student knowledge in concepts as more and more assessments are attempted.

4.4 Final Predictions

The final predictions for both, correct attempt and hint-taking, probabilities are calculated by applying two separate linear transformations followed by a sigmoid activation on $\mathbf{f}_t$ which is given by

$$\mathbf{f}_t = \text{Tanh}(\mathbf{W}_f^T * (\mathbf{s}_t || \mathbf{k}_t) + \mathbf{b}_f) \qquad (4)$$

Here, $\mathbf{W}_f$ is a linear transformation, $\mathbf{s}_t$ is the final read knowledge state of the student in question $\mathbf{q}_t$ illustrated earlier in Equation 3, $\mathbf{k}_t$ is the question embedding in Equation 1, $\mathbf{b}_f$ is the bias and $||$ is the concatenation operator. The final probabilities for a correct-attempt and hint-taking are

$$p_r^{\text{pred}} = \text{Sigmoid}(\mathbf{W}_r^T * \mathbf{f}_t + \mathbf{b}_p^r) \qquad (5)$$

$$p_h^{\text{pred}} = \text{Sigmoid}(\mathbf{W}_h^T * \mathbf{f}_t + \mathbf{b}_p^h) \qquad (6)$$

where both $\mathbf{W}_r^T$, $\mathbf{W}_h^T$ are linear transformations, and $\mathbf{b}_p^r$, $\mathbf{b}_p^h$ are bias vectors.

4.4.1 Prediction Loss at Output Layer:

The output layer of DKVMN predicts the probability whether a question will be answered correctly. For the task of predicting whether a hint will be taken in the question, the factors like the knowledge state of the learner, the difficulty level of the question and past hint-taking behavior are important. Since the first two are already being modeled by DKVMN, we learn both the tasks simultaneously by using a multi-task learning approach. As shown in Equation 6, the final output layer of Colearn adds a linear transformation of $\mathbf{f}_t$ followed by a sigmoid activation to predict the hint-taking task. The loss is given by taking a weighted sum of losses from knowledge tracing and hint-taking prediction and is evaluated as

$$\mathcal{L} = \alpha_1 \text{cross\_entropy}(p_r^{\text{act}}, p_r^{\text{pred}}) + \alpha_2 \text{cross\_entropy}(p_h^{\text{act}}, p_h^{\text{pred}})$$

where p_r^{pred} is given in Equation 5 and p_h^{pred} in Equation 6 are the probabilities predicted at the output layer. The actual values p_r^{act} and p_h^{act} are 0 or 1 depending on the observed response. The cross entropy function

$$\text{cross\_entropy}(p^{\text{act}}, p^{\text{pred}}) = p^{\text{act}}\log(p^{\text{pred}}) + (1-p^{\text{act}})\log(1-p^{\text{pred}})$$

We set both $\alpha_1 = \alpha_2 = 1$ to give equal weight to the knowledge tracing and hint-taking prediction tasks. This loss is backpropagated to update the network weights. When a learner takes a hint, only the loss of the hint-taking prediction is propagated. In other words, the loss for the knowledge tracing task is 0 in this case. The network weights, except the final output layer, are shared between the two tasks (See Figure 3). Multi-task learning acts as a regularizer for learning network weights as with the same set of weights the network should maximize two objectives. It also encourages sharing of knowledge across tasks through sharing of network weights. Experimental results demonstrate that the network trained using multi-task learning marginally outperforms current state-of-the-art models on both the tasks.

5. DATASETS

To evaluate the performance of the model we used the following two datasets:

- **ASSISTments 2009-2010 skill builder dataset**[2]: ASSISTments [14] is an online tutoring system which can be used by teachers for grade school-level Mathematics instruction and evaluation. The system can be used to identify common wrong answers and see student-reports for assignments in a class. The dataset contains activity logs of students solving exercises on the system and it is widely-used as a benchmark dataset for knowledge tracing [23, 34]. Log data includes information such as student responses, time spent on exercise, chronological order of attempts, if a hint is taken, tagged skill for an exercise. We use the updated version of this dataset. It corrects an issue, identified by Xiong *et al.* [33], with duplicated rows in the original version. We use the skill tag corresponding to an exercise as its identifier in the input to the models. Thus, the set of distinct questions, Q, is same as the set of distinct skill tags in the dataset. All rows with an empty skill tag are removed. Some rows contain invalid values in the column specifying student's first action i.e. values other than the permissible ones – {attempt, hint}. These transactions are removed. In case a student has multiple actions on the same exercise, we know whether the first action was a correct attempt, an incorrect attempt or a hint request. For the hint-taking prediction task, only the rows with the first action as a hint request are taken as a positive label.

- **Junyi Academy Math Practicing Log**[3]: Junyi Academy[4] is an e-learning platform, like Khan Academy, where students can practice exercises on various subjects including Mathematics, Biology, Computer Science. Like ASSISTments, the dataset contains attempt, hint taken, time spent, and skill tag information for an exercise. It has transactions for around 200,000 students. To the best of our knowledge, it is one of the largest student interaction datasets. As part of the data cleaning process, rows which contained non-binary values in the columns specifying whether hint was used or not and whether question

[2] ASSISTments 2009-2010 skill builder dataset is available at `https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data/skill-builder-data-2009-2010`

[3] Junyi Academy Math Practicing Log is available at `datashop.web.cmu.edu/DatasetInfo?datasetId=1198`

[4] `https://www.junyiacademy.org/`

was answered correctly or not were removed. Students with only one transaction in the dataset are removed. If a student requests a hint as one of the actions on a particular exercise, we do not know whether the hint was requested as the first action or it was requested after one or more incorrect attempts. In other words, we only know whether a hint request was one of the actions performed by the student. Therefore, for the hint-taking prediction task, all transactions which contain a hint request, irrespective of being the first action or not, are assigned the positive label. Note that this adds noise to the hint-taking label for this dataset.

The statistics comparing the two datasets are provided in Table 2.

Statistic	Datasets	
	ASSISTments	Junyi
# of Students	4,151	199,549
# of Exercise/Skill Tags	111	722
# of Concept Tags	–	40
# of Records	325,637	25,628,935
% of Attempts (Both Correct and Incorrect)	92.78%	93.56%
% of Hints	7.22%	6.44%

Table 2: Aggregate statistics from the two datasets

For extracting labels for the prediction tasks, it is assumed that a question is attempted only once. If a hint is taken first then the response is labeled as hint-taken. Else, the response is marked as correct or incorrect based on the outcome. So, if there are instances where multiple responses for a question are observed, we keep the first response on each question and remove subsequent responses. This is done to conform with the standard practice followed while evaluating knowledge tracing models. However, responses to subsequent attempts can also be incorporated in our setup.

6. EVALUATION METHODOLOGY

In each dataset, students and the corresponding transactions are randomly split into two parts – 80% for training and 20% for testing. Training set is further split, out of which 80% (i.e. 64% of total) is used for training the models. The rest 20% (i.e. 16% of total), called validation set, is used to tune hyperparameters of the models. Trained models with different values of hyperparameters are evaluated on the validation set in order to select the best hyperparameters.

6.1 Accuracy Metric

Both the prediction tasks are considered in a classification setting — answering a question correctly or not and taking a hint on a question or not. Hence, we compare the model performance based on Area under ROC curve (AUC) which is a standard classification metric. For knowledge tracing task, we follow the same evaluation procedure as followed by [23, 30, 34]. The model is trained using transactions from the training set. During the testing phase, the model is updated after each question response from the testing set. The updated model is used to perform the prediction for the next question.

6.2 Hyperparameter Tuning

Hyperparameters are learned using the validation set. We used Bayesian Optimization [28] to tune the hyperparameters for `Colearn` model. The model required several hyperparameters which cannot be set by hand easily. The method uses Bayesian techniques instead of gradient-based techniques to optimize the unknown function from the hyperparameter space to validation loss. The objective is to find the set of hyperparameter values which minimizes the validation loss while evaluating the model for only a small number of hyperparameter combinations. The tuned hyperparameters are:

Number of value vectors: Since the number of value vectors represent the number of 'hidden' concepts, this cannot be set by hand. The values were varied from 5 to 50 vectors.

Key vector size: The size of each key vector depends on efficient representation of the difficulty of questions and their similarity to the hidden concepts. The size was varied from 10 to 200.

Value vector size: The value vectors are a representation of the different concepts and an efficient representation depends on the size of these vectors. The size was varied from 10 to 200.

Hyper-parameters obtained for `Colearn` model are as follows – number of value vectors are 20 and 5 for ASSISTments and Junyi respectively, key vector size (i.e. question embedding size) is 50 for both, value vector size (i.e. question-attempt embedding size) is 200 and 100 for ASSISTments and Junyi respectively.

6.3 Training details

Stochastic gradient descent with momentum and norm-clipping was employed to train the weights of the network. The momentum was set to be 0.9 throughout the training and the norm was clipped to a threshold of 50.0. The learning rate was initialized as $5*10^{-2}$ and annealed after every 20 epochs till the learning rate reached 10^{-5}. Since the sequences of responses varied in length, the sequence length was fixed to 200 and 500 in ASSISTments and Junyi, respectively, with appropriate truncation or padding. Batch size for stochastic gradient descent is fixed to 32 and number of epochs is set to 100. Network weights corresponding to the epoch with least validation loss are taken for testing.

After training, learned weight values for the key and value matrices are saved and loaded at beginning of testing each student sequence. Key matrix is kept unchanged throughout the sequence, whereas the value matrix is updated independently for each student sequence as more actions are observed.

To check for robustness to initialization of network weights, we perform training 5 times with different random seeds (to get $\{\mathrm{AUC}_i\}_{i=1}^{5}$). We report the average (i.e. $\overline{\mathrm{AUC}} = \frac{1}{5}\sum_{i=1}^{5}\mathrm{AUC}_i$) and standard deviation (i.e. $[\frac{1}{5}\sum_{i=1}^{5}(\mathrm{AUC}_i - \overline{\mathrm{AUC}})^2]^{\frac{1}{2}}$) of test AUC values across the 5 models.

7. RESULTS AND DISCUSSION

Model	Datasets	
	ASSISTments	Junyi
Colearn	**91.75** ± 0.07%	**92.34** ± 0.009%
DKVMN–hints	91.12 ± 0.06%	92.31 ± 0.01%
HH (n=3)	77.69%	76.66%
HH (n=4)	79.10%	77.62%

Table 3: **Hint-taking Prediction task**. Performance (AUC values) of proposed approach (Colearn) compared with the baselines on two datasets.

Model	Datasets	
	ASSISTments	Junyi
Colearn	**81.48** ± 0.04%	**80.56** ± 0.009%
DKVMN	81.23 ± 0.02%	80.38 ± 0.007%
HIRT	77.40%	79.45%
IRT	76.51%	77.46%

Table 4: **Knowledge Tracing task**. Performance (AUC values) of proposed approach (Colearn) compared with the baselines on two datasets.

To the best of our knowledge, no prior work models both of the prediction tasks jointly. Therefore, we report comparisons with prior work for each task separately. The Colearn results reported are for the model jointly trained on both the tasks.

7.1 Hint-taking Prediction

7.1.1 Baselines

Castro *et al.* [9] proposed a method called Hint-History model (HH) for predicting student actions on next question i.e. whether student will take a hint or attempt the next question. The method considers the sequence of n most recent student actions for predicting action on the next question. They use a technique called *tabling method* which counts the number of times a sequence resulted in a particular action in the training set. For instance, while making a prediction for a student who has taken two hints in a row followed by an attempt, the method finds students with same action sequence in the training set and uses the next-action probability for them as the predicted value in current case i.e. calculate number of times students with this action sequence took hint on the next question divided by total number of such students in the training dataset. These simple approaches have been used for knowledge tracing tasks [13] as well.

The tabling method is compared with two approaches that are proposed in this paper. The first one is using DKVMN [34] model with class labels being hint-taking indicators instead of question correctness (referred to as DKVMN–hints). The second one is Colearn.

7.1.2 Results

Table 3 summarizes the results. We compare with HH model for two different values of length of action sequences, $n = 3, 4$. DKVMN–hints shows 12% points improvement in AUC on ASSISTments dataset and 15% on Junyi datset. Colearn further improves the AUC on the two datasets. A memory-augmented deep learning model considers longer term dependencies in student sequences instead of taking a fixed-length history, as is the case with HH. It can also effectively model student-specific variations from individual sequences whereas HH model output is based only on population-level statistics. Lastly, multi-task training, Colearn model, also helps to increase performance on the task by a small margin due to the synergies across the tasks.

7.2 Knowledge Tracing

7.2.1 Baselines

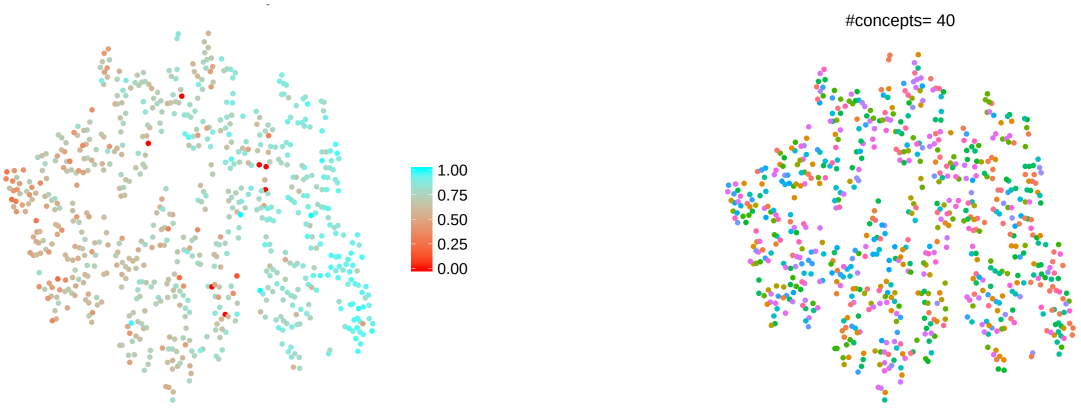
We compare our model with three competitive baselines namely DKVMN [34], IRT [30] and Hierarchical IRT (HIRT) [30]. In IRT, student skill level and item difficulty are modeled separately and probability of answering correctly is taken as a pre-determined function of these two quantities such as sigmoid or logistic. In HIRT, related items are grouped together (e.g. those belonging to same concept) and the difficulty of each item is distributed normally around a per-group mean, which is distributed normally around a hyper-prior. DKVMN model was shown to outperform BKT [11] and DKT [16], hence we do not compare with those models. For DKVMN, best performing hyperparameters reported in [34] were taken. Note that the best-reported AUC of DKVMN (81.57%) on ASSISTments dataset differs from what we report for their model (81.23%), for the same hyperparameters. This results from different train-test set proportions, i.e. 20% sequences in test as compared to 30% used by Zhang *et al.* We could replicate DKVMN results using code published by the authors[5] on the dataset split provided by them. For IRT and HIRT models we use the code published by the authors[6]. For the baselines, the transactions where hints are taken are labelled as incorrect responses. This is the same approach followed in the baseline publications.

7.2.2 Results

The AUC values for the different methods on both datasets for knowledge tracing are shown in Table 4. The AUC value for deep learning models is sensitive to the initial values of network weights. Hence, we report average and standard deviation (separated by ±) of the AUC from five, randomly initialized, models. Colearn improves test set AUC on ASSISTments dataset by 4% points and on Junyi by 1% points as compared to HIRT method. The improvement due to multi-task model is consistent across datasets and tasks, albeit small. This means that students' past hint taking behaviour is not predictive of question correctness. Factors such as difficulty of the question and correctness on past attempts mostly can explain their future performance. Interestingly, performance increase is less in case of Junyi dataset than ASSISTments dataset in both the tasks. As discussed earlier, the way hint information is available in Junyi dataset adds some noise to the training signals. In cases where student takes a hint, we do not know whether hint was the first action before any attempt or was taken after making incorrect attempt(s). This might be the reason why we get relatively less advantage from incorporating hint information in Junyi dataset.

[5] https://github.com/jennyzhang0215/DKVMN

[6] https://github.com/Knewton/edm2016



(a) Exercise Difficulty

(b) Exercise Concept Mapping

Figure 4: t-SNE visualizations of question representation for Junyi dataset. Color denotes difficulty (in (a)) and concepts (in (b)) of the questions.

7.3 Discussion on Learned Representations

We have shown that the `Colearn` model performs better than the baseline models. In this section we explore the meaning of the estimated parameters. Specifically, how can we use the estimated parameters to represent a question and what does the representation represent?

To get representation for each question, q_t, we use a question's attention weights over the concepts in the key matrix. Each question is represented by a vector of length equal to the number of latent concepts where the value corresponding to each latent concept in the vector is given by Equation 2. This representation is obtained assuming that a student has not yet started to answer any question. Recall that, before the start of an assessment, the value matrix is set to the initial value matrix, $\mathbf{M}_0^v$. This initial matrix is part of the parameter set and it is estimated. The question representation is a vector that is based on the performance of all students, questions, and responses in the training set but not specific to any one student.

To understand how the question representations are related to each other, we visualize them using t-SNE [18]. Figure 4a and Figure 4b present the t-SNE visualizations of the question representations of the exercise tags in Junyi dataset. ASSISTments dataset is not used for this analysis because it does not contain the concepts for the exercise tag. Each dot in the scatter diagram represents a single exercise tag. The only difference between the two panels is the color used to represent each tag. In Figure 4a each exercise tag is colored according to the difficulty level of the question, with blue color representing the easiest and red color representing the most difficult exercise tags. The difficulty level is estimated using the fraction of correct responses in each question tag. The color of a dot in Figure 4b represents the concept of the exercise tag. There are 40 concepts for 722 exercise tags in Junyi dataset which include concepts like *fractions*, *algebra*, *trigonometry*.

One of the hypothesis is that the question representation captures the concept map [34]. If this was the case then the exercise tags within a concept should be close in the question representation space. However, Figure 4b shows that the exercise tags within a concept do not cluster together. In fact, the exercise tags seem to be randomly scattered in the question representation space. On the other hand the color of the exercise tags in Figure 4a shows a definite pattern with the easiest question tags towards the left and the most difficult ones towards the right. This shows that the question representation vectors tend to capture the difficulty level of an exercise tag. Note that, the question representation vector might capture other aspects such as prerequisite map. However, a complete in-depth analysis is out of the scope of this paper and left for future explorations.

8. CONCLUSION

Assessments (specifically, formative ones) are an important part of an interactive learning system as they help learners to gauge their progress. If a learner is stuck at a particular question, many learning platforms provide learning aids in the form of hints. Predicting when to provide an option of taking an hint is essential to regulating its excessive use or to avoid underuse. The probability of taking a hint relates to modeling the knowledge state of a learner during an assessment, which has been studied separately as knowledge tracing. Hence, we jointly modeled the hint-taking prediction task along with the knowledge tracing task. Through experiments we showed that our approach outperforms the baseline hint-taking prediction models and marginally improve on baseline knowledge tracing models. The approach proposed in the paper can be easily extended to incorporate other types of learning aids such as interactive tutorials, links to reading material and videos.

Better knowledge tracing and hint-taking models allow an e-learning system to make decisions such as number of questions to ask, the sequence of questions and whether to show a hint based on learner's proficiency. Such decisions affect the long-term learning outcomes. Future work involves integrating the predictions for the two tasks to develop strategies for optimizing long-term learning outcomes. High accuracy on both the tasks, as demonstrated, will allow to build student simulators for evaluating such strategies.

9. REFERENCES

[1] V. Aleven and K. R. Koedinger. Limitations of student control: Do students know when they need help? In *Intelligent tutoring systems*, volume 1839, pages 292–303. Springer, 2000.

[2] V. Aleven, B. Mclaren, I. Roll, and K. Koedinger. Toward meta-cognitive tutoring: A model of help seeking with a cognitive tutor. *International Journal of Artificial Intelligence in Education*, 16(2):101–128, 2006.

[3] V. Aleven, E. Stahl, S. Schworm, F. Fischer, and R. Wallace. Help seeking and help design in interactive learning environments. *Review of educational research*, 73(3):277–320, 2003.

[4] V. A. Aleven and K. R. Koedinger. An effective metacognitive strategy: Learning by doing and explaining with a computer-based cognitive tutor. *Cognitive science*, 26(2):147–179, 2002.

[5] T. Bartholomé, E. Stahl, S. Pieschl, and R. Bromme. What matters in help-seeking? a study of help effectiveness and learner-related factors. *Computers in Human Behavior*, 22(1):113–129, 2006.

[6] J. D. Bransford and D. L. Schwartz. Chapter 3: Rethinking transfer: A simple proposal with multiple implications. *Review of research in education*, 24(1):61–100, 1999.

[7] A. Bunt, C. Conati, and K. Muldner. Scaffolding self-explanation to improve learning in exploratory learning environments. In *Intelligent Tutoring Systems*, pages 109–156. Springer, 2004.

[8] R. Caruana. Multitask learning. *Mach. Learn.*, 28(1):41–75, July 1997.

[9] F. E. V. Castro, S. Adjei, T. Colombo, and N. Heffernan. Building models to predict hint-or-attempt actions of students. *International Educational Data Mining Society*, 2015.

[10] C. Conati and K. Vanlehn. Toward computer-based support of meta-cognitive skills: A computational framework to coach self-explanation. *International Journal of Artificial Intelligence in Education (IJAIED)*, 11:389–415, 2000.

[11] 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.

[12] F. Drasgow and C. L. Hulin. Item response theory. *Handbook of industrial and organizational psychology*, 1:577–636, 1990.

[13] 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 *EDM*, 2013.

[14] M. Feng, N. Heffernan, and K. Koedinger. Addressing the assessment challenge with an online system that tutors as it assesses. *User Modeling and User-Adapted Interaction*, 19(3):243–266, 2009.

[15] S. A. Karabenick and J. R. Knapp. Help seeking and the need for academic assistance. *Journal of educational psychology*, 80(3):406, 1988.

[16] M. Khajah, R. V. Lindsey, and M. C. Mozer. How deep is knowledge tracing? *arXiv preprint arXiv:1604.02416*, 2016.

[17] X. Liu, J. Gao, X. He, L. Deng, K. Duh, and Y.-Y. Wang. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *HLT-NAACL*, pages 912–921, 2015.

[18] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[19] M. Mathews and T. Mitrović. *How Does Students' Help-Seeking Behaviour Affect Learning?*, pages 363–372. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[20] A. H. Miller, A. Fisch, J. Dodge, A. Karimi, A. Bordes, and J. Weston. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1400–1409, 2016.

[21] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

[22] A. Mitrovic. Self-explanation in a data normalization tutor. 2003.

[23] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 505–513, Cambridge, MA, USA, 2015. MIT Press.

[24] M. Puustinen. Help-seeking behavior in a problem-solving situation: Development of self-regulation. *European Journal of Psychology of education*, 13(2):271–282, 1998.

[25] A. Renkl. Worked-out examples: Instructional explanations support learning by self-explanations. *Learning and instruction*, 12(5):529–556, 2002.

[26] A. M. Ryan, M. H. Gheen, and C. Midgley. Why do some students avoid asking for help? an examination of the interplay among students' academic efficacy, teachers' social–emotional role, and the classroom goal structure. *Journal of educational psychology*, 90(3):528, 1998.

[27] S. Schworm and A. Renkl. Learning by solved example problems: Instructional explanations reduce self-explanation activity. In *Proceedings of the Cognitive Science Society*, volume 24, 2002.

[28] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[29] J. G. Trafton and S. B. Trickett. Note-taking for self-explanation and problem solving. *Human-computer interaction*, 16(1):1–38, 2001.

[30] K. H. Wilson, Y. Karklin, B. Han, and C. Ekanadham. Back to the basics: Bayesian extensions of irt outperform neural networks for proficiency estimation. *arXiv preprint arXiv:1604.02336*, 2016.

[31] D. Wood. Scaffolding, contingent tutoring, and

computer-supported learning. *International Journal of Artificial Intelligence in Education*, 12(3):280–293, 2001.

[32] H. Wood and D. Wood. Help seeking, learning and contingent tutoring. *Computers & Education*, 33(2):153–169, 1999.

[33] X. Xiong, S. Zhao, E. Van Inwegen, and J. Beck. Going deeper with deep knowledge tracing. In *EDM*, pages 545–550, 2016.

[34] J. Zhang, X. Shi, I. King, and D.-Y. Yeung. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th International Conference on World Wide Web*, pages 765–774. International World Wide Web Conferences Steering Committee, 2017.

[35] J. Zhou, J. Liu, V. A. Narayan, and J. Ye. Modeling disease progression via fused sparse group lasso. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1095–1103. ACM, 2012.

Job Description Mining to Understand Work-Integrated Learning

Shivangi Chopra and Lukasz Golab
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
{s9chopra,lgolab}@uwaterloo.ca

ABSTRACT

Work-integrated learning, also known as co-operative education, allows students to alternate between on-campus classes and off-campus work terms. This provides an enhanced learning experience for students and a talent pipeline for employers. We observe that co-operative job postings are a rich source of information about the required skills, working environment and company culture. We present a text mining methodology to extract and cluster informative terms from unstructured job descriptions, and we demonstrate the utility of our methodology on a co-op job posting corpus from a large North American university.

Keywords

work-integrated learning, co-operative education, text mining, Latent Semantic Analysis (LSA)

1. INTRODUCTION

The World Association for Cooperative and Work-integrated Education reports that 275 institutions from 37 countries offer co-operative education (co-op) programs, also referred to as work-integrated learning programs[1]. Students enrolled in co-op programs usually alternate between on-campus classes and off-campus work terms at participating employers. Co-operative education has become popular for a number of reasons: it provides an enhanced learning experience for students, a talent pipeline for employers, and a recruiting tool for institutions.

Concurrent with the popularity of work-integrated learning is the desire to understand the co-op job market: students want to know what types of jobs are available and what skills could make them more employable; employers want to know what competition they are facing and how to attract top talent; and institutions want to align curricula with job market needs.

In this paper, we propose to answer the above questions by mining co-operative job postings. We make two contributions: 1) a text mining methodology to extract informative terms from job descriptions in order to understand a co-op job market, and 2) a case study using real data to demonstrate our methodology.

In practice, job descriptions are written directly by employers, and therefore they are not standardized or well-structured. In particular, job descriptions may include information that is unrelated to the nature of the job such as website URLs, contact emails, and of course common English words. Our technical challenge, therefore, is to extract and cluster useful information, such as required skills, working environment and company culture.

We address this challenge by designing a text mining methodology to understand a co-op job market through job postings. We start by building a parser that extracts relevant attributes from unstructured job descriptions. We then identify frequently occurring attributes in job titles and descriptions, and we employ Latent Semantic Analysis (LSA) and k-means clustering over the extracted attributes to characterize the types of available jobs.

To demonstrate the utility of our methodology, we analyze nearly 30,000 co-op job postings from a large North American university. We identify sought-after skills and mindsets, we identify the types of jobs available to junior and senior undergraduate students, and we discuss trends over time. We argue that our findings provide actionable insights for students, employers and the institution.

The remainder of this paper is organized as follows. Section 2 discusses related work; Section 3 describes our data and methodology; Section 4 describes the experimental results; and Section 5 concludes the paper with the implications of our findings and directions for future work.

2. RELATED WORK

This paper is related to three bodies of work: text mining, co-operative education and workforce studies. We use standard parsing and information retrieval techniques, and do not make any new algorithmic contributions in text mining. Instead, our contribution is to apply these techniques to a new application domain in order to obtain new insight.

[1] http://www.waceinc.org/global_institutions.html

Prior work on co-operative education has focused on its impact on students' skills (especially soft skills such as leadership and entrepreneurship), grades and post-graduate employment; see, e.g., [2, 14, 21, 26, 29]. There has also been research on what makes co-op students successful and what workplace competencies are expected (see, e.g., [6, 7, 16, 20, 30, 31]), understanding competition for co-op jobs (see, e.g., [17, 27]), and assessing the overall co-op process and experience (see, e.g., [12, 18]). These works are orthogonal to ours, which studies a different problem of understanding a co-op job market in terms of the types of available jobs and the required skills and attitudes.

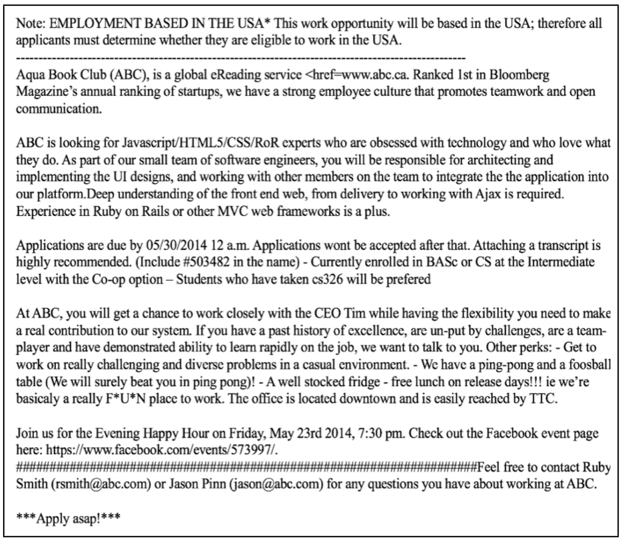
Prior research on job advertisements studied how to write them in order to attract qualified applicants (see, e.g., [4, 11, 22]), and how to match job descriptions with qualified resumes (see, e.g., [9, 19]). Moreover, job descriptions have been studied from a gender perspective, e.g., by counting the occurrences of masculine and feminine words [25]. While these works investigated how job descriptions could attract or match applicants, we study a different problem of understanding a co-op market through job descriptions.

Workforce literature has applied machine learning to improve recruitment, reduce turnover and understand work profiles [1, 5]. Machine learning algorithms have been applied to understand the factors affecting work performance and retention [5]. Furthermore, Aken et al. cluster Information Technology job postings on job websites to understand the work profiles prevalent in the market [1]. Our research extends this analysis to understand the work profiles of various industries (not only Information Technology) in a cooperative education setup and how they have changed over time. Not limiting the scope to broad work profiles, our research also highlights the specific skills and attitudes required by various industries.

3. DATA AND METHODOLOGY

We obtained two datasets from a large undergraduate North American institution: 12,066 job postings corresponding to all co-op jobs that were advertised and filled in 2004, and 17,057 job postings corresponding to all co-op jobs that were advertised and filled in 2014. The job postings are written in English. Most of these positions were located in North America, with a small number of overseas jobs. We use the 2014 data to characterize the current co-op job market and we compare with the 2004 data to analyze trends over time. Each record in our datasets contains the following information:

- A job title, up to 50 characters long, which generally consists of the position and/or the nature of the work. Common titles include Web Developer, Engineering Intern and Planning Assistant.
- A job description, with unlimited length and no standardized structure or formatting.
- The year of study of the successful candidate who secured the job. We refer to jobs obtained by first and second year students as *junior jobs* or *lower-year jobs*, and those obtained by third and fourth year students as *senior jobs* or *upper-year jobs*.



Note: EMPLOYMENT BASED IN THE USA* This work opportunity will be based in the USA; therefore all applicants must determine whether they are eligible to work in the USA.

Aqua Book Club (ABC), is a global eReading service <href=www.abc.ca. Ranked 1st in Bloomberg Magazine's annual ranking of startups, we have a strong employee culture that promotes teamwork and open communication.

ABC is looking for Javascript/HTML5/CSS/RoR experts who are obsessed with technology and who love what they do. As part of our small team of software engineers, you will be responsible for architecting and implementing the UI designs, and working with other members on the team to integrate the the application into our platform.Deep understanding of the front end web, from delivery to working with Ajax is required. Experience in Ruby on Rails or other MVC web frameworks is a plus.

Applications are due by 05/30/2014 12 a.m. Applications wont be accepted after that. Attaching a transcript is highly recommended. (Include #503482 in the name) - Currently enrolled in BASc or CS at the Intermediate level with the Co-op option – Students who have taken cs326 will be prefered

At ABC, you will get a chance to work closely with the CEO Tim while having the flexibility you need to make a real contribution to our system. If you have a past history of excellence, are un-put by challenges, are a team-player and have demonstrated ability to learn rapidly on the job, we want to talk to you. Other perks: - Get to work on really challenging and diverse problems in a casual environment. - We have a ping-pong and a foosball table (We will surely beat you in ping pong)! - A well stocked fridge - free lunch on release days!!! ie we're basicaly a really F*U*N place to work. The office is located downtown and is easily reached by TTC.

Join us for the Evening Happy Hour on Friday, May 23rd 2014, 7:30 pm. Check out the Facebook event page here: https://www.facebook.com/events/573997/.
##Feel free to contact Ruby Smith (rsmith@abc.com) or Jason Pinn (jason@abc.com) for any questions you have about working at ABC.

Apply asap!

Figure 1: An anonymized job description

- The academic program of the successful candidate.

Since the job postings in our dataset do not include industry or discipline labels, we use the academic program of the student who obtained the job as a proxy. The institution provided us with a mapping from students' academic programs to job disciplines; e.g., positions filled by Computer Science or Software Engineering students are classified as Information Technology jobs. In our case study, we focus on the largest discipline in the institution's co-op market: Information Technologies (IT). We also point out interesting findings from other major disciplines: Finance, Health Studies, Arts, Biology, Environmental Studies, Chemical Engineering, Civil Engineering, Electrical Engineering and Mechanical Engineering.

Figure 1 shows an anonymized example of a job description from our dataset. It includes the following information:

- Technical skills: Javascript, Ruby on Rails
- Soft skills: team player, ability to learn
- Job duties: architecting and implementing UI designs
- Desired mindset and attitude: obsessed with technology
- Perks: ping-pong and foosball table, free lunch
- Company culture: casual environment

However, there is also some content that does not describe the job itself: names of people and locations, URLs, email addresses, HTML tags, timestamps, special formatting, and, of course, common English words. The first part of our methodology, therefore, is a parser that extracts job-related attributes from unstructured job descriptions. The parser, implemented in Python, consists of the following steps.

1. Using regular expression matching, we remove URLs, HTML tags, phone numbers and other numbers, email addresses, timestamps, administrative annotations added by the institution (such as the text following "Note:" in Figure 1), formatting characters such as bullet points, and sequences of special characters serving as separators (such as the sequences of dashes and hashtags in Figure 1).

2. We tokenize the remaining text and remove special characters embedded in words (such as F*U*N in Figure 1). To remove unimportant terms, we build a vocabulary, called *Remove-List*, consisting of common English words[2], misspellings[3] and abbreviations[4], as well as manually-curated lists of company names, locations, addresses and persons' names appearing in the institution's co-op system.

3. We have to be careful to not remove informative terms. For example, "Ajax" is a city in Canada and is therefore in *Remove-List*. However, Ajax is also a Web development toolkit. To address this problem, we create another vocabulary called *Keep-List*, of words that should *not* be removed. This vocabulary consists of skills found on a resume help Web site[5] and job duties from the Canadian National Occupation Classification[6]. Note that *Keep-List* only contains a subset of words we are interested in; e.g., it is missing many specific technical skills, perks and company culture descriptors.

4. We stem the remaining tokens using the NLTK snowball stemmer [7] and we remove stop words. Finally, we leverage our domain knowledge by converting important terms that can be written in different ways into a standard form; e.g., "java-script" and "javascript" both map to "javascript".

At the end of the parsing process, each job description is reduced to its stemmed words, minus those in *Remove-List* but not in *Keep-List*. In the remainder of the paper, we will refer to these stemmed words as "words", "terms", "tokens" and "attributes" interchangeably.

The second part of our methodology is designed to analyze the extracted job attributes. We do this in two ways:

1. To identify popular skills, attitudes, working environment and perks, we report attributes that occur at least once in a large percentage of job descriptions. Notably, and in contrast to other text mining applications, we do not count the number of occurrences of an attribute within a posting–we observed that important job requirements such as knowledge of the "Java" programming language are usually mentioned only once. We also identify attributes mentioned by more junior than senior jobs (and vice versa), and we compare attributes mentioned by more jobs in 2014 than 2004 (and vice versa) to characterized trends over time.

2. We use clustering to identify the different types of available co-op jobs within a discipline. Following previous work on text clustering [10, 23, 24], we start by applying Latent Semantic Analysis (LSA) to the job descriptions, with each job description represented as a *job vector*. The ith coordinate of a job vector is equal to the inverse document frequency (IDF) of the ith word in the set of possible words, provided that this word is mentioned in the given job description at least once (and zero otherwise). Following previous work, we use LSA to reduce the dimensionality of job vectors from the number of distinct words down to one hundred [28]. Each reduced dimension corresponds to a latent concept in the data. We then run k-means clustering on the transformed job vectors, and we report a few top terms (again, ranked by IDF) from each cluster centroid as representatives.

Table 1: Top 10 frequent tokens in IT job titles

Token	Freq. in 2014	Token	Freq. in 2004
softwar	45%	develop	37%
develop	44%	softwar	27%
analyst	8%	analyst	17%
applic	7%	programm	11%
web	5%	assist	9%
support	4%	web	8%
assist	4%	support	7%
programm	4%	applic	6%
system	3%	system	6%
quality	3%	specialist	4%

4. RESULTS

In this section, we demonstrate the utility of our methodology. We show in-depth results for the largest discipline in our dataset: Information Technologies (IT), including frequent term analysis (Section 4.1), analysis of significant differences in term frequencies between 2014 and 2004 and between senior and junior jobs (Section 4.2), and clustering analysis (Section 4.3). We summarize our results for other disciplines in Section 4.4.

4.1 Frequent Term Analysis

Table 1 shows the top 10 attributes occurring in the most IT job *titles* in 2014 and 2004; for example, the first row indicates that the token "softwar" appears at least once in 45% of job titles in 2014 and 37% in 2004. Not surprisingly, nearly half the titles mention software development.

Table 2 shows the top 25 attributes occurring in the most IT job *descriptions* in 2014 and 2004. Overall, most IT co-op jobs appear to be software developer jobs. In 2014, hardware was mentioned in only 14% of the postings and embedded systems in 7%; in 2004, these percentages were slightly

[2] http://www.lextutor.ca/freq/lists_download/longman_3000_list.pdf

[3] https://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings/For_machines

[4] https://media.gcflearnfree.org/ctassets/modules/48/common_abbr.png

[5] https://www.thebalance.com/list-of-the-best-skills-for-resumes-2062422

[6] http://noc.esdc.gc.ca/English/noc/welcome.aspx?ver=16

[7] www.nltk.org/_modules/nltk/stem/snowball.html

Table 2: Top 25 frequent attributes in IT job descriptions

Token	Freq. in 2014	Token	Freq. in 2004
develop	91%	develop	80%
team	84%	applic	65%
softwar	76%	softwar	62%
applic	66%	system	61%
design	65%	team	61%
product	62%	program	54%
program	60%	design	53%
system	58%	communic	50%
project	53%	comput	49%
comput	52%	product	47%
test	50%	support	43%
build	48%	test	43%
communic	48%	servic	42%
web	47%	project	41%
code	46%	lead	39%
help	46%	excel	39%
learn	45%	solut	38%
servic	44%	web	38%
java	43%	tool	37%
manag	43%	assist	36%
creat	43%	busi	36%
solut	42%	manag	35%
technic	42%	java	35%
tool	41%	custom	34%
excel	40%	oper	33%

higher, at 22 and 9, respectively (and the actual number of hardware and embedded systems jobs was slightly higher in 2004). Furthermore, about half the job descriptions mention testing. Notably, mentions of some soft skills such as communication are more frequent than mentions of specific technical skills such as Java in both years.

By inspecting other frequent attributes, we obtain the following insights about frequently mentioned programming languages, platforms and applications in 2014:

- Programming languages: Java (43%), C++ (33%), JavaScript (31%), C (24%), Python (22%), C# (20%), HTML (19%), CSS (17%), PHP (12%), .NET (12%), jQuery (10%), Perl (10%), XML (9%), Ruby (9%)
- Development: web (47%), mobile (32%), game (12%)
- Databases: database (29%), SQL (26%), mySQL (8%), Oracle (7%)
- Mobile applications: android (19%), iPhone (7%)
- Operating Systems: linux (21%), unix (13%), iOS (14%)
- User-centered development: user (35%), agile (18%), deploy (16%)
- Other applications: server (29%), distributed (17%), security (17%), cloud (9%), graphic development (8%), big data (4%)
- Concepts: OOP (Object-Orient Programming) (24%), algorithms (18%), scalable (14%)

In terms of the working environment and company culture, the strongest result is that the word "team" is very frequent, suggesting a collaborative environment. Other frequent terms include challenging (32%), dynamic (20%), fun (16%), flexible (15%) and diverse (12%). Amenities such as free food, foosball and ping-pong tables are also frequent. The word start-up is mentioned in 11% of the job postings.

We also note the occurrence of mindset-related terms such as learn (45%), innovation (32%), passion (25%), focus (23%), creativity (22%), motivation (20%), love (15%) and enjoy (10%).

Similarly, for 2004, we identify the following frequently mentioned programming languages, platforms and applications:

- Programming languages: Java (35%), C++ (31%), C (21%), HTML (22%), XML (15%), ASP.NET (12%), Perl (11%), .NET (10%), JavaScript (10%), JSP (8%), C# (7%)
- Development: web (38%), mobile (10%), game (5%)
- Databases: database (30%), SQL (27%), Oracle (13%), mySQL (2%)
- Operating Systems: unix (22%), linux (15%)
- User-centered development: user (21%), deploy (7%), agile (0.5%)
- Other applications: server (25%), security (15%), graphic development (10%)
- Concepts: OOP (13%), algorithms (7%), scalable (4%)

Compared to 2014, the word "team" was again frequent in 2004, but words related to mindset, company culture and perks were less frequent.

Our results indicate that IT positions focus on software rather than hardware, especially web and Java development. The work environment appears team-oriented. In 2014, descriptions of mindset and company culture are appearing frequently.

4.2 Significant Differences

Next, we investigate the differences between 2014 and 2004 IT job descriptions which we began to see in the previous section. Table 3 summarizes the results by listing 20 attributes with most significant differences in frequencies between 2004 and 2014 (on the left), and 2014 and 2004 (on the right). We define a difference in frequencies, abbreviated Δ, as the percentage of job postings mentioning an attribute in one year minus the percentage of job postings mentioning this attribute in the other year. Both lists are sorted by Δ, and all results shown are statistically significant with P-values less than 0.05 using a proportion test [13]. We omit the analysis of job title differences between 2004 and 2014 which gave similar results. We also show a Venn diagram in Figure 2, which illustrates the overlap among the top 100 frequent attributes in 2004 and 2014 IT jobs.

Table 3: Differences in frequency between job description attributes of 2004 and 2014 IT

Token	2004	2014	Δ	Token	2014	2004	Δ
assist	36%	22%	14%	build	48%	15%	33%
asp	12%	2%	10%	help	46%	19%	26%
internet	18%	9%	9%	team	84%	61%	24%
unix	22%	13%	8%	code	46%	24%	22%
hardwar	22%	14%	8%	mobil	32%	10%	22%
sort	8%	0%	8%	javascript	31%	10%	21%
clarifi	8%	1%	8%	passion	25%	5%	20%
interperson	18%	10%	7%	featur	30%	10%	20%
oper	33%	26%	7%	creat	43%	23%	20%
msaccess	8%	1%	7%	python	22%	3%	19%
manufactur	10%	4%	6%	learn	45%	26%	19%
cost	11%	5%	6%	collabor	23%	5%	18%
xml	15%	9%	6%	agil	18%	0%	18%
support	43%	37%	6%	product	62%	47%	16%
expens	8%	2%	6%	contribut	27%	12%	15%
intranet	7%	1%	6%	problem	34%	19%	15%
oracl	13%	7%	5%	improv	25%	10%	15%
prepar	11%	6%	5%	solv	20%	6%	15%
supervis	12%	7%	5%	app	15%	1%	14%
xp	8%	3%	5%	peopl	33%	18%	14%

Table 4: Differences in frequency between job description attributes of junior and senior jobs in 2014 IT

Token	Jr.	Sr.	Δ	Token	Sr.	Jr.	Δ
document	29%	16%	13%	c++	46%	21%	24%
support	42%	31%	11%	algorithm	28%	9%	20%
assist	27%	16%	11%	scale	28%	9%	19%
communic	53%	43%	10%	scienc	49%	31%	17%
manag	48%	38%	10%	featur	39%	22%	17%
test	54%	45%	9%	python	31%	14%	16%
report	26%	17%	9%	scalabl	23%	7%	16%
busi	42%	34%	9%	build	57%	41%	15%
written	21%	13%	8%	code	54%	40%	15%
activ	23%	15%	8%	complex	27%	13%	13%
educ	17%	10%	7%	comput	59%	46%	13%
standard	15%	8%	7%	c	31%	18%	13%
interperson	13%	6%	7%	product	69%	57%	13%
instal	9%	3%	7%	structur	21%	9%	12%
troubleshoot	15%	9%	6%	field	23%	11%	12%
msoffic	8%	2%	6%	java	50%	38%	12%
summari	24%	18%	6%	data	42%	30%	12%
execut	15%	9%	6%	distribut	23%	12%	11%
detail	11%	5%	6%	search	16%	6%	10%
account	12%	6%	6%	problem	40%	29%	10%

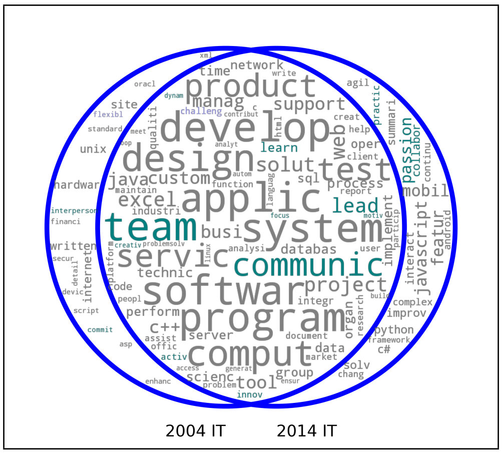


Figure 2: Overlap between the top 100 most frequent attributes of IT jobs in 2004 and 2014

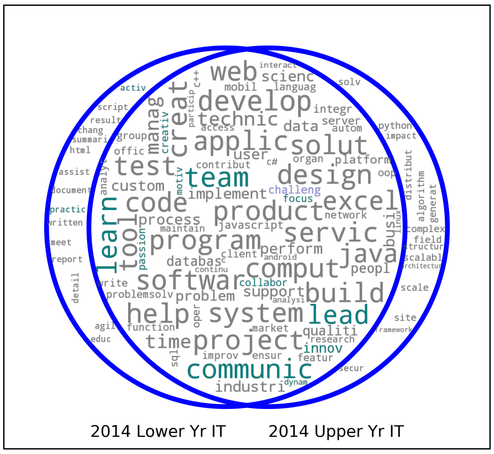


Figure 3: Overlap between the top 100 most frequent attributes of Junior and Senior IT jobs in 2014

Our results suggest that 2004 job postings include more entry-level positions (suggested by attributes such as "assist", "support", "prepare", "arrange" and "document"), and mention technologies and software popular at the time such as ASP, XML, Windows XP and Microsoft Access. Additionally, the fraction of hardware-oriented jobs was higher in 2004. On the other hand, job postings in 2014 include words representing current technologies such as mobile, Javascript, Python, agile and app (and, further in the list, scalable and distributed systems). Notably, many soft skills and mindset-related terms are more frequent in 2014: "passion", "create", "learn", "collaborate" and "contribute". Although not shown in Table 3, other terms that are more frequent in 2014 include company culture descriptors such as "innovative","challenging", "fun" and "diverse".

The next important difference is that between junior and senior jobs. Table 4 shows two lists: top terms appearing in more junior than senior jobs (on the left), and top terms appearing in more senior than junior jobs (on the right), both in 2014 and both sorted by the difference of percentages. Table 5 shows the same two lists, but for 2004. Figures 3 and 4 show Venn diagrams that illustrate the overlap among the top 100 frequent terms from junior and senior jobs in 2014 and 2004, respectively.

We observe that in 2014, junior jobs are more likely to be entry-level documentation, testing or troubleshooting jobs. Junior job postings are more likely to mention soft skills such as communication and interpersonal skills. In terms of specific technologies, junior jobs mention HTML, SQL and Web 5 percent more frequently than senior jobs. On the other hand, senior jobs in 2014 mention technical concepts and specific programming languages such as algorithms, scalability, data, C++, C and Python. Other interesting differences not shown in the table are OOP (9% more frequent

Table 5: Differences in frequency between job description attributes of junior and senior jobs in 2004 IT

Token	Jr.	Sr.	Δ	Token	Sr.	Jr.	Δ
maintain	23%	14%	9%	c++	45%	21%	24%
support	47%	38%	9%	c	32%	14%	18%
updat	13%	5%	8%	design	63%	46%	17%
html	26%	18%	8%	cost	20%	5%	14%
excel	43%	35%	8%	clarifi	16%	2%	14%
msoffic	13%	5%	8%	expens	16%	2%	14%
troubleshoot	14%	7%	8%	arrang	17%	4%	13%
document	30%	23%	7%	sort	16%	3%	13%
user	24%	17%	7%	solut	44%	33%	11%
qualiti	26%	20%	6%	challeng	29%	19%	10%
report	26%	20%	6%	develop	85%	76%	9%
web	40%	34%	6%	linux	20%	11%	9%
mainten	14%	8%	6%	complex	16%	7%	9%
instal	13%	7%	6%	algorithm	12%	3%	9%
interperson	20%	15%	5%	unix	27%	18%	9%
server	27%	22%	5%	code	29%	21%	8%
hardwar	24%	19%	5%	lead	44%	36%	8%
xp	10%	5%	5%	innov	27%	19%	8%
time	31%	26%	5%	oop	18%	10%	8%
offic	21%	17%	5%	scale	11%	3%	8%

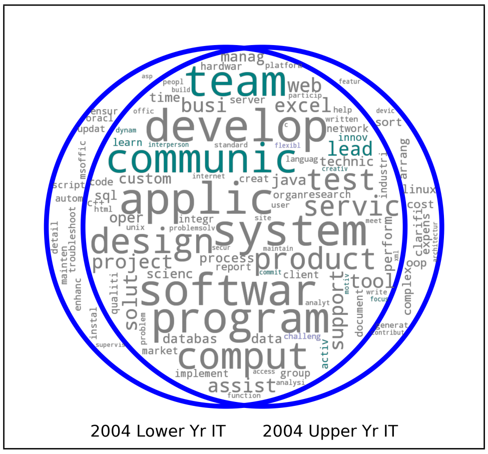


Figure 4: Overlap between the top 100 most frequent attributes of Junior and Senior IT jobs in 2004

than in junior jobs), linux (8%), cloud (8%), security (7%) and data science (5%).

We observe similar patterns in Table 5 and Figure 4. In 2004, junior jobs also included terms suggesting entry-level positions, whereas senior jobs included more mentions of programming languages and computing concepts.

To summarize, there are clear differences between 2014 and 2004 IT jobs, and between junior and senior jobs. In addition to differences due to new technologies, soft skills, mindset and company culture are more frequently mentioned in 2014. In both years, junior IT jobs are more likely to mention documentation, testing and troubleshooting, whereas senior jobs are more likely to mention technical concepts.

Table 6: Largest clusters of 2014 IT jobs

Label	Tokens in cluster centroid	%All	%Jr.	%Sr.
Web Development	javascript, html, web, css, sql, c#, server, java, net, jquery	22%	**64%**	36%
Programming	c++, c, languag, linux, python, oop, scienc, algorithm, perl, script	21%	46%	**54%**
Start-up Culture	startup, python, javascript, featur, code, web, love, stack, fun, passion	18%	39%	**61%**
Business Analyst	sql, analyst, test, solut, c#, script, execut, financi, document, busi	16%	**69%**	31%
Mobile Development	io, android, mobil, app, platform, java, agil, iphon, devic, c	10%	**61%**	39%
System Administrator	hardwar, troubleshoot, configur, instal, network, desktop, server, user, xp, resolut	6%	**87%**	13%

4.3 Clustering Analysis

After investigating frequently occurring terms, we now cluster the IT job descriptions to understand the types of available jobs. We experimented with different numbers of clusters between 2 and 30. We present results using ten clusters; using fewer clusters led to different types of jobs being assigned to the same cluster, whereas using more clusters led to similar types of jobs belonging to multiple clusters.

Table 6 shows the six largest clusters in 2014 sorted by size; the remaining four clusters had under 2% of the total number of jobs each. We report the representative tokens of each cluster centroid, a manually-assigned label summarizing the tokens, and three percentages: the percentage of all jobs assigned to this cluster, and the percentages of junior and senior jobs within this cluster. We highlight the higher of the last two percentages in bold font to indicate whether a cluster consists of more junior or senior jobs.

Based on the clustering results, we characterize the IT co-op market as follows. The five largest clusters cover 87% of IT jobs, spanning web development (22%), programming (21%), start-ups (18%), business analysis (16%) and mobile development (10%). The junior vs. senior split evident in the clustering is consistent with our earlier results from Section 4.2: troubleshooting jobs are mostly filled by junior students, whereas jobs mentioning company culture, many of which are startups, are filled by senior students.

Table 7 shows the 7 largest IT clusters in 2004; the remaining three clusters are small and one of them contains job postings from a specific large employer at the time. There is no longer a start-up cluster with mentions of the working environment, and there is an emphasis on hardware in the last cluster. These results align with our earlier results from Section 4.1.

To summarize, our clustering methodology segments the IT job market into web development jobs, general software development jobs, data analysis jobs, mobile

Table 7: Largest clusters of 2004 IT jobs

Label	Tokens in cluster centroid	%All	%Jr.	%Sr.
Software Development and Testing	java, test, sql, tool, server, qualiti, softwar, autom, custom, solut	20%	**61%**	39%
Web Development	html, sql, web, asp, javascript, server, java, xml, databas, net	19%	**66%**	34%
Databases	scienc, databas, model, comput, analysi, group, research, data, tool, msaccess	15%	**57%**	43%
System Development	c++, sort, clarifi, expens, arrang, cost, gui, code, java, softwar	15%	40%	**60%**
System Administrator	network, hardwar, troubleshoot, instal, user, configur, xp, desktop, msoffic, problem	9%	**87%**	13%
Programming	perl, script, languag, unix, c, java, rank, c++, enterpris, linux	7%	**57%**	43%
Embedded Systems and Graphics	video, digit, hardwar, c, multimedia, debug, embed, devic, c++, graphic	7%	36%	**64%**

development jobs and troubleshooting jobs. Mentions of mindset and work environments in 2014 are frequent enough to create a separate cluster for these jobs.

4.4 Analysis of Other Disciplines

In this section, we apply our text mining methodology to the other disciplines in our dataset. As before, we structure the results into frequent term analysis, difference analysis (2014 vs. 2004 and junior vs. senior jobs), and clustering analysis to characterize the types of available jobs within each discipline. We focus on job description analysis and only mention the results of job title analysis if they lead to additional insight.

4.4.1 Frequent Term Analysis

Overall, all the other disciplines have frequent mentions of soft skills ("team", "communication", "leadership") and basic computing skills (databases and Microsoft Office) in both 2004 and 2014. Below, we highlight additional frequent terms for each discipline.

Finance: soft skills indicating client relationships ("client", "interpersonal", "relationship"); finance-specific technical skills ("audit", "tax", risk assessment, asset valuation, market analysis); formal office working environment ("bank", "office")

Health Studies: soft skills ("active students", indicating physical fitness); health-specific terms ("patient", "care", "kinesiology", "therapy", "injury", "rehabilitation", "ergonomics", "physiotherapy", "recreation")

Arts: tokens related to editorial, technical and content writing ("edit", "write", "english", "proofread", "content"); additionally, media and social media were frequently mentioned in 2014.

Biology: discipline-specific technical terms ("molecular", "chemistry", "microbiology", "biochemistry", "disease", "cell", "tissue", "DNA", "genetics", "pharmaceutical"); lab-oriented work environment ("research", "lab", "technician")

Environmental Studies: discipline-specific terms (GIS (Geographic Information System), "water", "land", "soil", "map", "survey", "sample", "policy"); field work environment ("field", "site"). Frequent words in job titles: "assistant", "planner", "technician", "research", "analyst", "inspector", "project", "management".

Chemical Engineering: Discipline-specific technical terms ("chemistry", "process", "manufacturing", "equipment", "sample", "procedure", process improvement, "safety"); lab-oriented work environment (chemical plants, research labs). Additionally, frequent in 2014: project management; frequent in 2004: field-work.

Civil Engineering: construction-related tokens ("design", "AutoCAD", "site", "field", "concrete", "safety"); graphic design ("graphic", "PhotoShop").

Electrical Engineering: discipline-specific technical skills ("electrical", "hardware", "power", "schematic", "control", "embedded", "circuit"); computing skills ("code", Web, Java, SQL). Frequent terms in job titles: "design", "quality", "assurance", "testing", "research".

Mechanical Engineering: discipline-specific terms ("equipment", "assembly", "robot", "circuit", "material", "CAD", "SolidWorks", "AutoCAD", "control", "process", "improvement", "maintenance", "draw", "prototype", "test", "troubleshoot", "safety"); work environment ("plant", "shop", "floor", "manufacturing").

4.4.2 Significant Differences

Next, we highlight differences in frequent terms between 2004 and 2014. Overall, we observed that each discipline had more mentions of soft skills, and more mentions of project management and IT-related terms in 2014. Additional differences are summarized below for each discipline.

Finance: 2004 jobs mention actuarial science more; 2014 jobs mention risk management and assessment, "equity", "trade", "client" and "interaction" more. Additionally, 2014 jobs mention concepts related to data analysis (e.g., Microsoft Excel and VBA).

Health Studies: 2014 jobs include more research related terms: "research", "summary", "data", "review", "cancer". 2004 jobs have more mentions of "recreation", "kinesiology", "outdoor", "therapy" and "teach". In particular, "cancer" appears in 6% more job postings in 2014 than in 2004.

Arts: more 2014 jobs mention market analysis and media-related terms: "media", "project", "management", "PowerPoint", "client" and "relationship". 2004 jobs mention more writing-related terms such as "history", "newsletter", "proofread", "French" and HTML.

Biology: 2014 job postings include more research and project management positions, and mention computing skills and clinic more often. 2004 job postings mention laboratory terms including "technique", "microbiology", "sample", "gel", "biochemistry", "microbe", HPLC (High Performance Liquid Chromatography blood test), "bacteria".

Environmental Studies: 2014 jobs mention project management, clients, research and computing skills more often. 2004 jobs mention "educate", "air", "waste", "treatment", "recycle" and ground water. It is interesting to note that "sustain" (sustainability) is mentioned 7% more often in 2014 than in 2004.

Chemical Engineering: 2014 jobs mention project management terms (e.g., "manage", "report", "project", "maintain"), "safety", "energy", "oil", "gas", "petroleum" and "sand" more often than 2004 jobs. On the other hand, 2004 jobs mention more computing skills and laboratory-specific terms ("lab", "technician", "sample", "treatment").

Civil Engineering: 2014 jobs mention more software ("software" and "AutoCAD" are mentioned 21% and 8% more often, respectively, in 2014 than in 2004). 2004 jobs mention "cost" and "expense" more often than 2014 jobs. It is interesting to note that "safety" is mentioned 13% more often in 2014 than in 2004.

Electrical Engineering: 2014 jobs mention "passion" and computing skills related to web development, core programming languages and mobile development. 2004 jobs mention more "manufacturing", "graphic", "multimedia", "processor", "hardware", "VHDL" (a hardware description language) and "Unix".

Mechanical Engineering: 2014 jobs mention research (suggested by "lab", "research", "simulate", "electron"), client-oriented development ("client", "customize") and computing terms (Python, Java, "mobile"). 2004 jobs are more likely to mention mechanical engineering terms: "blueprint", "draw", "cost", "weld", "hydraulics", "gear". It is interesting to note that "quality" is mentioned 9% more in 2014 than in 2004. While both AutoCAD and SolidWorks are CAD software, SolidWorks is mentioned 11% more in 2014 while AutoCAD is mentioned 5% more in 2004.

Next, we compare the differences between tokens in junior and senior jobs in each discipline. Overall, more senior jobs across all disciplines mention project management or deal with advanced concepts of the field (either through applications or research). Junior jobs appear to have more clerical work, computing-related responsibilities or mention less advanced concepts of the discipline (including testing, field work and lab work). We provide additional discipline-specific details below.

Finance: Senior jobs require more technical knowledge of the field ("audit", "invest", "risk" "management"). Junior jobs have a more clerical ("document", "arrange", "English") and computing (HTML, Java, databases) focus. Senior jobs are more likely to mention "commitment", "dynamism", "client" and "interaction". Additionally, senior jobs in 2004 mention more mathematical and statistical terms than junior jobs in 2004.

Health Studies: Senior jobs mention more research. Junior jobs mention more field work.

Arts: Senior jobs mention more project management (suggested by "manage", "PowerPoint", "client", "workload", "process", "improvement"). Junior jobs mention more clerical work, "English", "Web", "research" and "customer service". Additionally, senior jobs in 2004 appear to include more business analyst and editor roles than junior jobs in 2004.

Biology: Senior jobs mention more "research", "hospital" and technical terms including "genetics", "therapy", "cancer", "cardiovascular", "nanomedicine", "biomaterial" and "in vitro". Junior jobs are more likely to mention "office", "assistant", "support" and "campaign".

Environmental Studies: Senior job titles indicate more planner and analyst positions with more project management, policy-making and GIS terms mentioned in the descriptions. Junior job titles indicate more lab technician, inspector, and surveyor positions with more "lab", "survey", "test" and "outdoor" mentioned in the descriptions. 2004 senior jobs additionally mention environmental concepts including "ground", "water", "remedy", "contaminate", "river", "hydrology" and "hydrogeology".

Chemical Engineering: Senior jobs mention a more industrial working environment with more mentions of "energy", "product", "design", "cost", "process", "improvement" and "optimization". Junior jobs mention laboratory-specific terms ("research", "sample", "record") more often. In 2004, senior jobs mentioned more chemical manufacturing terms.

Civil Engineering: Senior jobs mention more "modelling", "design", "client", "interaction" and "software". Junior jobs mention more "inspection", "field", "survey", data recording and clerical work. 2014 senior jobs have more mentions of project management.

Electrical Engineering: Senior jobs mention more electrical concepts ("power", "circuit", "embedded", "distributed", PCB (Printed Circuit Board), "sensor", "chip", "schematic"). Junior jobs mention more quality assurance and basic computing terms ("web", "program") as well as more clerical work. In addition, junior jobs in 2004 contain system administrator positions and senior jobs in 2004 mention more programming languages (C++ and C).

Mechanical Engineering: Senior jobs are more likely to mention project management, designing and implementation. Junior jobs have more clerical (e.g., "update", "arrange", "email", "written"), computing (marked by "database", "Web", SQL, HTML, Java) and field-work, and requirement collection terms ("client", "custom", "meet"). Junior jobs in 2004 do not mention client interaction; instead they mention testing.

4.4.3 Clustering Analysis

Finally, we apply our clustering methodology to each discipline, both for 2014 and 2004. Our clustering results provide additional support for the findings in Section 4.4.1 and 4.4.2. Additionally, the main benefit of clustering is that it reveals the different types of available jobs in each discipline. We discuss these findings below.

Finance: In 2014, the largest clusters were: several clusters mentioning finance-specific skills such as "trade", "equity", "tax", "reconciliation", "pension", asset valuation, risk management, "forecast", "causality" and "insurance" (63%); financial documentation (15%); and Web software development (10%). The jobs clustered under finance-specific skills were dominated by senior students, with the clerical (documentation) and IT (web development) clusters dominated by junior students. This result aligns with our analysis of significant differences from the previous section. Furthermore, in 2004, the largest clusters relate to financial analysis and documentation (51%), actuarial work including "valuation" and "pension" (18%), "tax" and "audit" (14%), and "causality" and "insurance" (5%). Thus, the 2004 clusters focus more on documentation and appear to describe a narrower range of jobs. All clusters except the last one mentioned have an equal split of junior and senior jobs.

Health Studies: The largest clusters in 2014 are related to organizing community events (21%), recreation camps for adults and children (14%) and therapy (13%), and are dominated by junior jobs. Smaller clusters dominated by senior jobs are related to research, cancer patient care and advanced aspects of health studies, including biomechanics, anatomy and statistics. The 10 clusters in 2004 are similar but exhibit equal proportions of junior and senior jobs in recreation, leisure and patient care.

Arts: The largest job clusters in 2014 include writing online content (24%), organizing events and providing customer service (22%), and writing, proofreading and summarizing research material (13%). These clusters have an almost even split of junior and senior jobs. Other clusters include project management (indicated by "stakeholder", "PowerPoint", "present"), market analysis ("campaign", "blog", "promote"), content writing ("Drupal", WCMS, standing for Web Content Management System), library liaisons and teaching (adult education, names of courses), which are dominated by senior students. Additionally, 52% of the jobs in 2004 fall in one cluster characterized by preparing English material for education and research on various topics including policy and politics. Other clusters include publishing newsletters and articles (with "graphics") (12%), office assistant positions (indicated by words such as "multitask", "file", "compile", "photocopy" and "fax") (8%), teaching and business analysis. Most of the clusters have an almost even split between junior and senior jobs. It appears that the Internet and social media have created new Arts jobs.

Biology: Our clustering results identify jobs in various fields of this discipline (microbiology, molecular biology, genetics, biochemistry), using various techniques (chromatography, electrophoresis).

Environmental Studies: The largest clusters in 2014 include project management (31%), education/research (25%), survey (18%), urban planning (13%) and advanced topics including GIS, cartography and geospatial analysis. (13%). On the other hand, half the jobs in 2004 mention educating people (largest cluster). While 8% of the jobs are related to advanced concepts, the other three clusters involve urban planning (20%), hydrogeology (14%) and waste water treatment (12%).

Chemical Engineering: Clustering 2014 Chemical jobs reveals additional insight: there is a cluster of jobs related to mechanical aspects of chemical plants, including the term "equipment". Additionally, a cluster with "nanotechnology", "lab", "material" and "physics" includes 10% of 2014 jobs. While 8% of the jobs are related to energy sources (including "oil", "gas", "petroleum", "sand" and "biofuel"), 5% of the jobs revolve around "emission", "environment", "pollution", "regulation" and greenhouse gases. Similar to 2014, 2004 clustering also contains clusters related to the mechanics of chemical plants, process improvement and research. It is interesting to note the differences in the field of application in both the years. While 2014 concentrates on nanotechnology, energy and emissions, 2004 deals with pharmaceuticals and waste water treatment.

Civil Engineering: Consistent with the previous section, junior students dominate the clusters including on-site field work (data collection and inspection), and senior students dominate the design clusters.

Electrical Engineering: The types of jobs in 2014 include System development (18%), web development (14%), electrical drawing (12%), PCB and circuit design (12%), system administration (9%), quality assurance (9%), simulation/research (8%), power (8%), embedded systems (8%) and research on advanced topics including transmitters, effect on climate, power grids, etc. (2%). In line with the findings of the previous section, there is a higher proportion of junior jobs in computing and system administration, and a higher proportion of senior jobs in core electrical clusters including circuit design and embedded systems. The main types of jobs in 2004 are related to power systems (26%), IT (19%), project management (18%), circuit design (15%), multimedia/graphics (6%), and transmission/telecommunication (4%).

Mechanical Engineering: Three-quarters of both 2004 and 2014 Mechanical Engineering jobs fall in the mechanical drawing cluster. While the other quarter of 2004 jobs mention plant-related terms including "assembly", "weld" and "motor", the other quarter of 2014 jobs is related to computing ("hardware", "automate", C++, Java, C, "web", "code"). Clustering 2014 jobs further reveals a 60-20-20 split among mechanical drawing, embedded systems and web development jobs.

To summarize, our clustering methodology identifies the types of available jobs in various disciplines. Through frequent term analysis, we found that soft skills and basic computing skills appear to be important in all disciplines in the 2014 job dataset.

5. DISCUSSION AND CONCLUSIONS

In this paper, we presented a text mining methodology to extract, compare and cluster important terms from freetext job descriptions. Our method identifies required skills as well as working environment and company culture descriptors. To demonstrate the utility of our methodology, we analyzed a dataset containing nearly 30,000 undergraduate co-operative job postings from two years: 2004 and 2014. Our main findings are as follows.

- As expected in an undergraduate co-op marketplace, there are many assistant and junior positions, but less so in 2014 than in 2004.
- Basic computing skills are needed in almost all disciplines and at all levels. In other words, many non-IT disciplines appear to be trending towards IT.
- Soft skills are mentioned frequently by job postings from all disciplines, and more so in 2014 than in 2004. For example, over all disciplines, "team" was mentioned 20% more often in 2014 than in 2004. (in 71% vs. 51% of all job postings). These findings agree with those reported in [3, 8, 15]. Besides teamwork, communication and leadership were frequently mentioned in job postings across all disciplines, with IT postings additionally mentioning mindset-related terms (passion and love for the work), Finance jobs mentioning interpersonal relationships and Health Studies jobs recruiting active students.
- Regardless of discipline, lower-year positions were and are more clerical and/or involve more basic computing. Upper year positions tend to mention advanced concepts and solution methods.
- We identified several trends over time by comparing 2004 jobs with 2014 jobs. For example, IT jobs now emphasize mobile and cloud computing, Arts jobs involve social media and Chemical Engineering jobs mention sustainable energy.
- Job postings from different disciplines suggest different working environments: plants in Chemical and Mechanical Engineering, labs in Biology, and casual, fun and collaborative environments in IT.

We emphasize that our results should be interpreted carefully due to the following factors.

- Diversity in size and age of companies, e.g., the IT discipline has many modern companies that emphasize a fun work culture, while other disciplines such as Finance have more traditional companies which might emphasize client relationships.
- Incorrect job descriptions which may not reflect the true nature of the job; e.g., employers may write or modify the job descriptions to suit the company's public image.

Nevertheless, we believe that our findings are of interest to students, employers and the institution. We provide several examples of actionable insights below.

- We can provide students with a better understanding of co-op opportunities in various disciplines and therefore help them select the right academic program and career.
- In particular, we suggest that all students, regardless of discipline, acquire basic computer programming skills, which should help them secure co-op positions in their junior years.
- The institution can use our findings to manage the expectations of junior students. As we showed, it may take until senior years to obtain a co-op position that fully utilizes advanced discipline-specific skills.
- The institution may use frequently appearing job attributes and the clustering of jobs in various disciplines to produce more effective promotional material for its co-op programs and to help attract strong students.
- With the help of our findings, the institution can make an informed decision about how to change academic curricula to align with employers' needs. For example, as all disciplines seem to emphasize teamwork, the institution can incorporate more team exercises in the curriculum. Hackathons and other competitions could be organized to foster passion and other mindset-related skills for IT students, while mock client meetings could be arranged for Finance students so that they could hone their interpersonal skills. New tools and methods may be introduced in courses when the corresponding terms begin to appear in job descriptions.
- Employers may examine our findings to understand which skills are in high demand and to understand the extent of competition in the co-op market.
- Our lists of frequent attributes may be used to redesign the way employers submit job postings. For instance, separate fields (outside the job description) may be added for required skills and company culture descriptions, with drop-down lists populated with frequent terms obtained through our methodology. Additionally, our clustering methodology can be used to segment the job descriptions to make it easier for students to find jobs they are interested in.

Naturally, there is more data-driven work that can be done. The goal of a successful co-op system is to match the right student with the right employer. Thus, our long-term research objective is to minimize the gap between employers' needs and students' talents. In this paper, we focused on job descriptions, which provide an indication of what co-op employers are looking for and what working environments they offer. In future work, we will characterize what students have to offer by mining resumes. Furthermore, we plan to study the gap between what employers want and what is being taught in schools (e.g., by comparing job postings with course descriptions). Another interesting research direction is to determine if students are likely to obtain full-time jobs at one of their co-op employers after graduating. Finally, we are interested in comparing our job postings with those from other institutions worldwide. For example, the knowledge of foreign languages did not appear to be important in our dataset but it may be important in other countries.

6. REFERENCES

[1] A. Aken, C. Litecky, A. Ahmad, and J. Nelson (2010). Mining for computing jobs. IEEE Software, 27(1):78-85.

[2] A. Andrade, S. Chopra, B. Nurlybayev and L. Golab (2018). Quantifying the impact of entrepreneurship on co-operative job creation. International Journal of Work-Integrated Learning, 19(1):51-68.

[3] R. Bancino and C. Zevalkink (2007). Soft skills: The new curriculum for hard-core technical professionals. Techniques: Connecting Education and Careers (J1), 82(5):20-22.

[4] A. E. Barber and M. V. Roehling (1993). Job postings and the decision to interview: A verbal protocol analysis. Journal of Applied Psychology, 78(5):845-856.

[5] C. F. Chien, and L. F. Chen (2008). Data mining to improve personnel selection and enhance human capital: A case study in high-technology industry. Expert Systems with Applications, 34(1):280-290.

[6] R. K. Coll, K. E. Zegwaard and D. Hodges (2002). Science and technology stakeholders ranking of graduate competencies part 1: Employer perspective. Asia-Pacific Journal of Cooperative Education, 3(2):19-28.

[7] R. K. Coll, K. E. Zegwaard and D. Hodges (2002). Science and technology stakeholders ranking of graduate competencies part 2: Students perspective. Asia-Pacific Journal of Cooperative Education, 3(2):35-44.

[8] R. De Villiers (2010). The incorporation of soft skills into accounting curricula: preparing accounting graduates for their unpredictable futures. Meditari Accountancy Research, 18(2):1-22.

[9] M. Diaby, E. Viennet, and T. Launay (2013). Toward the next generation of recruitment tools: an online social network-based job recommender system. In Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 821-828.

[10] C. Ding and X. He (2004). K-means clustering via principal component analysis. In Proceedings of the 21st International Conference on Machine Learning, 29.

[11] D. C. Feldman, W. O. Bearden and D. M. Hardesty (2006). Varying the content of job advertisements: The effects of message specificity. Journal of Advertising, 35(1):123-141.

[12] S. Ferns and K. Moore (2012). Assessing student outcomes in fieldwork placements: An overview of current practice. Asia-Pacific Journal of Cooperative Education, 13(4):207-224.

[13] J. L. Fleiss, B. Levin and M. C. Paik (2004). Determining Sample Sizes Needed to Detect a Difference between Two Proportions. Statistical Methods for Rates and Proportions, pages 64-85. John Wiley & Sons, Inc.

[14] J. Gault, J. Redington and T. Schlager (2000). Undergraduate business internships and career success: are they related? Journal of marketing education, 22(1):45-53.

[15] I. Grugulis and S. Vincent (2009). Whose skill is it anyway? Soft skills and polarization. Work, employment and society, 23(4):597-615.

[16] D. Hodges and N. Burchell (2003). Business graduate competencies: Employers views on importance and performance. Asia-Pacific Journal of Cooperative Education, 4(2):16-22.

[17] Y. Jiang and L. Golab (2016). On Competition for Undergraduate Co-op Placements: A Graph Mining Approach. In Proceedings of the 9th International Conference on Educational Data Mining (EDM), 394-399.

[18] Y. Jiang, S. W. Y. Lee and L. Golab (2015). Analyzing student and employer satisfaction with cooperative education through multiple data sources. Asia-Pacific Journal of Cooperative Education, 16(4):225-240.

[19] E. Malherbe, M. Diaby, M. Cataldi, E. Viennet, and M. A. Aufaure (2014). Field selection for job categorization and recommendation to social network users. In Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 588-595.

[20] E. Rainsbury, D. L. Hodges, N. Burchell and M. C. Lay (2002). Ranking workplace competencies: Student and graduate perceptions. Asia-Pacific Journal of Cooperative Education, 3(2):35-44.

[21] E. Ralph, K. Walker and R. Wimmer (2009). Practicum-education experiences: Post-interns' views. International Journal of Engineering Education, 25(1):122-130.

[22] C. L. Reeve and L. Schultz (2004). Job-seeker reactions to selection process information in job ads. International Journal of Selection and Assessment, 12(4):343-355.

[23] W. Song and S. C. Park (2007). A novel document clustering model based on latent semantic analysis. In Proceedings of the International Conference on Semantics, Knowledge and Grid, 539-542.

[24] S. E. Sorour, T. Mine, K. Goda, and S. Hirokawa. (2014). Efficiency of LSA and K-means in Predicting Students' Academic Performance Based on Their Comments Data. In Proceedings of the International Conference on Computer Supported Education (CSEDU), Volume 1, 63-74.

[25] S. Tang, X. Zhang, J. Cryan, M. J. Metzger, H. Zheng, and B. Y. Zhao (2017). Gender Bias in the Job Market: A Longitudinal Analysis. In Proceedings of the ACM on Human-Computer Interaction, 1(CSCW), article 99.

[26] G. R. Thiel and N. T. Hartley (1997). Cooperative education: A natural synergy between business and academia. SAM Advanced Management Journal, 62(3):19.

[27] A. Toulis and L. Golab (2017). Graph Mining to Characterize Competition for Employment. In Proceedings of the Network Data Analytics workshop at the ACM SIGMOD Conf. on Management of Data, 3:1-3:7.

[28] TruncatedSVD [Computer software manual]. Accessed on 6 March 2018, at `scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html`

[29] D. R. Young, D. N. Stengel, P. Chaffe-Stengel and R. M. Harper (2010). Assessing the academic and

workplace skills of undergraduate business interns. Journal of Cooperative Education and Internship, 44(1):13-22.

[30] K. E. Zegwaard and D. Hodges (2003). Science and technology stakeholders' ranking of graduate competencies part 3: Graduate perspective. Asia-Pacific Journal of Cooperative Education, 4(2):23-35.

[31] K. E. Zegwaard and D. Hodges (2003). Science and technology stakeholders' ranking of graduate competencies part 4: Faculty perspective. Asia-Pacific Journal of Cooperative Education, 4(2):36-48.

Gender Differences in Undergraduate Engineering Applicants: A Text Mining Approach

Shivangi Chopra, Hannah Gautreau, Abeer Khan, Melicaalsadat Mirsafian and Lukasz Golab
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
{s9chopra,hvgautre,a383khan,mmirsafi,lgolab}@uwaterloo.ca

ABSTRACT

It is well known that post-secondary science and engineering programs attract fewer female students. In this paper, we analyze gender differences through text mining of over 30,000 applications to the engineering faculty of a large North American university. We use syntactic and semantic analysis methods to highlight differences in motivation, interests and background. Our analysis leads to three main findings. First, female applicants demonstrate a wider breadth of experience, whereas male applicants put a greater emphasis on technical depth. Second, more female applicants demonstrate a greater desire to serve society. Third, female applicants are more likely to mention personal influences for studying engineering.

Keywords

Gender differences, engineering, admissions, text mining, clustering.

1. INTRODUCTION

The failure of science and engineering programs to attract equal numbers of women and men is well-documented; only 23% of women with high scores in mathematics pursue Science, Technology, Engineering and Mathematics (STEM) degrees as compared to 45% of men with the same scores [9]. As a result, there has been a great deal of research on understanding why this is the case; see, e.g., [1, 3, 4, 13, 18, 19, 20]. The major findings of prior work are that women are less likely to pursue STEM degrees because they do not see how this leads to societal improvement, and that women are more often led to study engineering because of influences from family and friends. Prior work has also found that the gender gap in STEM fields is *not* due to a difference in technical ability.

One weakness of existing work is that it is based on small datasets collected through surveys and longitudinal studies. In this paper, we present a large-scale text mining study of this topic. Our analysis is enabled by a unique dataset of over 30,000 undergraduate applications to the engineering faculty of a large North American university. Applicants are required to describe why they are interested in studying engineering, and provide other relevant information such as reading interests, extracurricular activities and programming experience. Our goal is to *determine whether female applicants identify different reasons for applying to an engineering program, and whether female applicants have different technical and extracurricular backgrounds.*

To answer these questions, we use text mining to extract the *reasons* why students apply to engineering programs. As in other text mining applications, challenges arise due to the ambiguity of natural language. To overcome these challenges, we rely on word embeddings and clustering to partition the text into semantically meaningful groups. We also analyze gender differences in programming languages and extracurricular activities through classification models and word frequency analyses. To the best of our knowledge, there is no prior work on large-scale text mining to obtain insights about students' motivation and interests.

The main findings of this paper are that women differentiate themselves through breadth of experience and men differentiate themselves through technical depth; women more often display a desire to serve society; and that women are more likely to mention interpersonal relationships when discussing their engineering goals.

The remainder of this paper is organized as follows. Section 2 summarizes related work; Section 3 discusses our dataset and methodology; Section 4 presents our results; Section 5 discusses the implications of our findings; and Section 6 concludes the paper with directions for future work.

2. RELATED WORK

There are three areas of work on gender differences in STEM. First are qualitative studies on small populations of students through interviews and surveys. Second are statistical studies that use census data or other summary data. Third, there are data mining studies on student performance. These works span students who are in high school, already enrolled in STEM programs, and who are working in a STEM profession.

First, we discuss qualitative survey-based studies.

Diekman et al. [3] studied 360 students from STEM and non-STEM fields consisting of 57.5% women. Each participant was asked about their mathematics and science experience and their perception of the degree to which different careers fulfill their personal goals. Participants' answers reflected that STEM careers impede communal-goal endorsement, which refers to how much a field enables achieving the goal of helping people and society. It was found that gender can predict communal-goal endorsement, and that communal-goal endorsement can negatively predict interest in STEM and positively predict interest in female-dominated programs with higher accuracy than other metrics such as gender or self-efficacy. Eccles [4] found similar results on a larger, more comprehensive dataset. They presented a longitudinal study of 1500 participants from south eastern Michigan from 6th grade to adulthood. They found that the main source of gender differences in entry to STEM careers is not gender differences in mathematical ability, but differences in inclinations towards society-oriented jobs. Women who aspire to math-related or engineering careers place a lower value on society-oriented job characteristics than their female colleagues who did not aspire to STEM careers.

Matusovich et al. [13] examined gender differences in values, but only within engineering. The study was conducted on 6 women and 5 men who majored in engineering. Each student was interviewed once a year throughout their undergraduate degree, and asked how his or her values affect their decision to earn an engineering degree. Values were classified under 4 groups: Attainment (ability to see oneself as an engineer), Cost (time and effort involved in their studies), Interest (enjoyment of understanding how math and science can be applied to every day life), and Utility (potential for future earnings). It was found that women were less likely to see themselves as engineers but continued to pursue an engineering degree due to the other values.

More reasons to pursue engineering were observed by Smith [19]. Smith interviewed 17 women who were studying engineering at four different colleges in the United States. Smith observed that participants were influenced to study engineering by family or friends. These influences played a pivotal role in helping the women build self confidence in their mathematical and science ability. They found an expression of "love" towards mathematics in many cases, despite the fact that these courses were also considered difficult. An interest in physics was found to be instrumental in their decision to study engineering. Women chose engineering because it allowed them to utilize the concepts covered in physics without having to major in physics. However, gender differences were not considered.

In terms of quantitative studies based on summary statistics, Hango [9] found that while mathematical ability plays a role, it does not explain gender differences in STEM career choices. Women with high mathematical ability are less likely to enter STEM fields than even men with a lower mathematical ability. He also supported the findings of Eccles suggesting that the gender gap in STEM programs is due to other factors.

There is prior work on gender differences in STEM using data mining techniques [16, 5, 10, 12]. However, these findings focus on student performance, whereas our work focuses on students' motivations for studying STEM, and their non-academic experiences and backgrounds.

Finally, there exists work on gender differences in computing, but it focuses on attitudes toward computing and proficiency with basic tasks [20, 1, 18]. Instead, we focus on reported programming language knowledge.

To the best of our knowledge, our work is the first one that conducts a data driven analysis of the reasons why students want to pursue engineering, and calculates the gender differences in these reasons. We also study past employment experiences, and programming knowledge in an effort to capture a more holistic view of the personalities of women and men who apply to engineering. In our conclusions, we verify some of the results of previous studies, and add to others.

3. DATA AND METHODOLOGY

3.1 Data

Our dataset comes from the engineering faculty of a large North American university. It contains all applications – both accepted and rejected – to the 14 available engineering programs from 2013 to 2016 inclusive. Table 1 shows the number of applications and the gender distribution of the applicants to each program, sorted by percentage of female students. The dataset includes gender, first choice program, and short free text responses to the following fields:

1. Engineering interests and goals: explain why you are interested in engineering and the specific program to which you applied.
2. Reading interests: discuss a book or an article you enjoyed or that has had an impact on you (preferably something that was not part of a course at school).
3. List any extracurricular activities or areas of significant interest.
4. List any jobs you held throughout high school.
5. Only mandatory for applicants to Software Engineering: list any programming experience you have.
6. Additional information: tell us anything else about yourself that you would like us to know when we review your application.

We report results for three groups of applicants: Biomedical and Environmental Engineering (BEE), Software Engineering (SE) and all other programs (OTHER). We initially analyzed applications to each program separately but observed applicants to programs within OTHER to be similar in the trends they display. Notably, the gender split in BEE is equitable, unlike other programs which are male-dominated. Furthermore, SE has unique application requirements (programming knowledge) and requires additional analysis.

Table 1: Gender breakdown by program

Program	Applicants	% Women	% Men
Environmental	1021	53%	47%
Biomedical	2015	52%	48%
Chemical	3612	38%	62%
System Design	957	38%	62%
Management	1040	36%	64%
Civil	3375	28%	72%
Geological	361	25%	75%
Nanotechnology	1670	24%	76%
Electrical	3782	17%	83%
Computer	3931	16%	84%
Software	3635	14%	86%
Mechanical	5473	12%	88%
Mechatronics	2886	12%	88%
Total	33758	23%	77%

3.2 Methodology

We use *syntactic* and *semantic* methods to analyze the free text responses. Syntactic methods identify words mentioned by more men or women, or words that can predict gender. Additionally, we apply semantic methods to "Engineering Interests and Goals" to capture context and extract the reasons why men and women want to study engineering.

3.2.1 Syntactic Analysis

For each of the six free text fields, we first perform standard pre-processing: we remove stop words, tokenize the text, and stem the tokens using the NLTK snowball stemmer[1]. We then perform two syntactic analyses on each field:

Document Frequencies: we identify words used at least once by a larger fraction of men or women (where each response is considered a document). We only report statistically significant differences with a P-value of 0.05 using a proportion test [6].

Gender Prediction: we build classifiers to predict gender based on the words or contiguous sequences of words (bigrams and trigrams) appearing in a free text response. Following previous work on text classification, we use logistic regression [8] where the dependent variable is gender, and the explanatory variables correspond to the possible words (or word bigrams/trigrams), and their values correspond to their TF-IDF scores [15, 21]. To calculate a TF-IDF score for a given word and a given response, we divide the number of times the word appears in the response by the Inverse Document Frequency - the fraction of responses in the entire dataset containing this word. TF-IDF is a useful measure because it balances the uniqueness of a term in the corpus and the importance of the term to the specific document. For each free text field except programming experience, we report the F-measure, which is the weighted harmonic mean of precision and recall [2], and accuracy, both calculated using 10-fold cross validation. We use oversampling for SE and OTHER to control for gender imbalance; otherwise, a classifier that always predicts gender as "male" would have a high accuracy on any male-dominated dataset.

Table 2: Families of programming languages

Family	Constituent Programming Languages
Java	java, bluej, jython, android
C++	c++, beta
Python	python
HTML/CSS	html, html5, css, css3
C	C, objective-C, robotc
JavaScript	javascript, jscript, jquery, angularjs
Turing	turing, touring
C#	c#, visual c#
Php	php
SQL	sql, pl/sql
Other	.net, ada, alice, applescript, bash, etc

3.2.2 Analysis of Programming Experience

In the "Programming Experience" field, SE applicants are asked to list their programming experience. The structure of this question elicits not only specific programming languages, but also encourages applicants to share details about their programming experience. Thus, in addition to the document frequency analysis mentioned earlier, we perform the following detailed analyses:

- Programming Language analysis: we calculate the number of responses that mention a given programming language. We start with a list of known languages from Wikipedia[2]. We then add common misspellings of these languages, and we group them into families in consultation with a domain expert. Table 2 shows the language families whose frequencies we will report.
- Programming Concept analysis: we compile a list of computing concepts, a sample of which is shown in Table 3, group them into categories, and calculate the number of responses that mention a given concept.
- Learning Method analysis: we compile a list of online programming courses, and common variations of "high school", "self taught", "higher education", and "employment". We then categorize these terms according to how an applicant learned programming: "online", "high school", "self taught", "higher education", "work", and "other". Finally, we calculate the number of responses that mentioned each learning method.
- Experience analysis: we extract the amount of experience reported by an applicant by searching for the words "hour", "day", "month", "year", as well as common abbreviations and misspellings of these words. We use the token immediately preceding these words to determine the length of time. We convert all of the times into months.

3.2.3 Semantic Analysis of Engineering Interests

Using the responses to "Engineering Interests and Goals", we want to identify the reasons why students apply to engineering programs. However, reasons cannot be inferred

[1] http://www.nltk.org/_modules/nltk/stem/snowball.html

[2] https://en.wikipedia.org/wiki/List_of_programming_languages

Table 3: Sample of programming concepts

Concept Category	Constituent Concepts
Basic	array, list, loop, if-statement
Data Structures	stack, queue, linked list
Sorting	merge sort, bubble sort, quick sort
Searching	linear, binary, breadth first searches
OOP	object, class, abstraction, encapsulation
Data Science	machine learning, NLP
Other	storage, memory management

Table 4: Nine questions used with the QA API

Question No.	Question Variant
1	Why are you interested in Engineering?
2	What inspired you to study Engineering?
3	What do you find inspiring about Engineering?
4	What are the reasons you like Engineering?
5	Why do you feel the need to pursue Engineering?
6	Why are you passionate about Engineering?
7	Why does Engineering interest you?
8	Why do you want to study Engineering?
9	Why do you like Engineering?

simply by counting occurrences of certain keywords; for example, family influence may be expressed by using words such as “father”, “mother”, “uncle”, or simply, “family”. Furthermore, an applicant may mention things other than the exact reason as to why they are interested in engineering in their response. Our semantic approach deals with these issues through the use of *Question Answering* to isolate topics being mentioned that could be considered indicative of reasons, followed by *Clustering using Word Embeddings* to analyze these. Figure 1 shows the steps in our semantic analysis, and they are explained in detail below.

1. Question Answering (QA): Here, we extract sentences that are most likely to contain the topics indicative of the applicants' underlying reasons for applying to engineering. We use a state of the art QA network [17] which is available as an open source API[3]. Given a question and a text document, this QA API extracts a sentence that may answer the question. However, we discovered that while asking the question that directly appeared on the entrance application - why are you interested in engineering - yielded *some* relevant sentences, there were additional relevant sentences that were not identified. To address this problem, we consulted with domain experts at the institution and formulated additional variants of this question. Depending on the applicant, not every variant identified a unique sentence. Overall, we observed that the number of unique sentences extracted per applicant plateaued at nine question variants. Table 4 lists the nine variants we use and Table 5 gives an example of the sentences extracted from a particular response using each question.

2. Stop Word Removal: Next, we remove stop words from the sentences extracted in the previous step because these do not contain any meaningful information about the underlying reasons. Similarly, we remove words excessively used by both genders such as “engineering” and the name of the university. This step happens after QA because QA requires the complete text, stop words included, as input.

Table 5: Sentences extracted from a particular response using all 9 question variants

Question No.	Answer produced by the QA API
1	future entrepreneurship ventures
2	designing & building complicated solutions
3	future entrepreneurship ventures
4	intellectual curiosity and satisfaction is core to my personality
5	i think i fit in well in the tight culture of the engineering class
6	intellectual curiosity and satisfaction is core to my personality
8	intellectual curiosity and satisfaction is core to my personality
7	know people much closer
9	it's the best program available

3. Sentence Vector Computation: At this point, each response has produced up to nine relevant sentences. We use *word embeddings* to capture semantic proximity between sentences. Specifically, we use the word2vec model [14], trained on the Google news corpus[4], to convert each word into a 300-dimensional vector that encodes the underlying semantics. We then use the average of all word vectors in a sentence as its *sentence vector*. If two sentence vectors are close, the sentences are also semantically similar [7, 11].

4. Clustering of Sentence Vectors: Next, we cluster the sentence vectors received from the previous step using K-Means clustering with Euclidean distance as the similarity metric and $K = 200$, where K is the number of clusters (the rationale behind this choice of K will be discussed shortly). The clusters converge around similar topics. For example, sentences containing words related to family such as “brother”, “father”, or “sister” have similar word vectors and are more likely to be assigned to the same cluster. Note that this would not be the case had we clustered the sentences themselves according to their *syntactic* similarity.

5. Cluster Representative Extraction: After computing clusters of sentence vectors, we extract representative words from each cluster to identify the topic of that cluster. First, we map sentence vectors back to the original sentences, which creates 200 sets of sentences, one set for each cluster. We then tokenize and stem the text in each set, as described in Section 3.2.1. The word2vec model consumes unstemmed words, compelling us to postpone tokenization and stemming until this step. The trigrams in each set are ranked using their TF-IDF scores calculated considering all 200 sets as the corpus. Finally, we represent each cluster with a list of 10 top ranking trigrams, an example of which is shown in Table 6.

[3] https://github.com/allenai/bi-att-flow

[4] https://code.google.com/archive/p/word2vec/

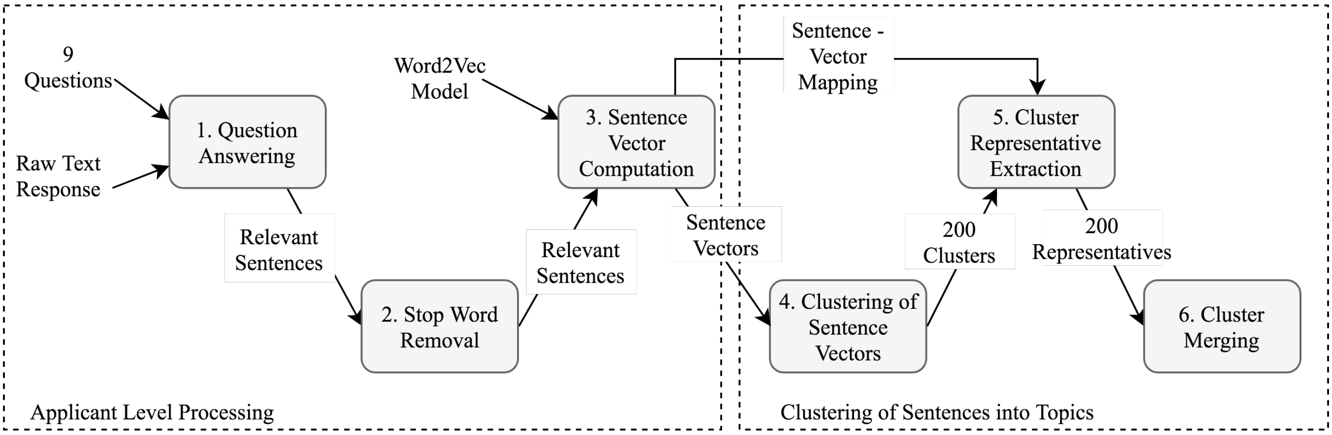


Figure 1: Semantic analysis methodology

Table 6: An example of ten trigrams representing a cluster

Rank	Trigram
1	solv problem solv
2	problem solv problem
3	enjoy problem solv
4	problem solv enjoy
5	enjoy solv problem
6	love solv problem
7	problem solv love
8	love problem solv
9	problem problem solv
10	solv problem problem

Table 7: Examples of a mixed cluster using $K = 50$ and its pure equivalent using $K = 200$

Mixed Cluster (50 Clusters)	Pure Cluster (200 Clusters)
kid work day	kid work day
apart tri togeth	young age father
love thing apart	visit construct site
countless hour spent	watch father work
use everi day	older brother mechan
pay close attent	dad electr took
decid high school	work day dad
work day saw	like help father
day day basi	expos young age
year high school	uncl civil engin

Choice of K: In **Step 4**, we experimented with values of K ranging from 50 to 200. When choosing a small K and proceeding to **Step 5** with fewer clusters, many clusters were represented by trigrams that were not semantically similar enough to warrant being in the same cluster, and some uncommon trigrams were overpowered by extremely common ones. Thus, some nuanced topics were lost as they could not form a cluster of their own. Larger values of K resulted in the splitting out of semantically similar topics. These resulted in pure but redundant clusters, i.e., several clusters representing the same topic. For instance, Table 7 shows a cluster of mixed topics on the left when K is 50, and a rather pure cluster on the right when K is 200. A bigger K made it possible for topics like "kid work day" to be grouped with similar semantic contexts like "watch father work". The topics on the right consistently speak of the influence of a family member, indicative of family influence as a reason for engineering, whereas no single reason can be deduced from the cluster on the left. The first K value that produced adequately pure clusters was 200. Thus, the decision was made to stop testing larger values and creating further unnecessary redundancy. To eliminate the unnecessary redundancies at $K = 200$, the clusters were merged in **Step 6**.

6. Cluster Merging: At this point we have 200 clusters of sentences, where each cluster is represented by the 10 highest ranking trigrams. To make the clusters interpretable and to group them under more general topics, we manually merge similar clusters based on their 10-trigram representations to produce ten final clusters. This process of merging follows the Card-sorting approach. Card-sorting has been widely used to systematically derive taxonomies from data, to reach a higher level of abstraction, and identify common themes [22]. For instance, it can be used to sort responses to an open-ended question into bins to deduce themes over the responses. We perform card-sorting on the representative trigrams, then we brand each of the ten final themes with human interpretable labels and consider these our final **topics**. In this process, a number of small clusters whose representatives were vague were disregarded, but even then, 99.5% of applicants were labelled with at least one topic. Table 8 shows two examples of representatives of vague clusters. Since the QA in **Step 1** used questions probing the reasons why the applicant was applying to engineering, our topics can be considered indicative of the same.

Table 9 shows the final set of topics along with sample trigram representations of clusters that were classified under each topic. *Technical Interests* refers to characteristics inherent to engineering along with topics related to specific engineering disciplines. For instance, the trigram "water treatment plant" in Table 9 is part of *Technical Interests* while being specifically related to Environmental Engineering.

All the sentences classified under a specific topic in Table 9 are tracked back to the applicants who mentioned them.

Table 8: Examples of discarded vague clusters

Example #1	Example #2
appli program appli	pursu decid pursu
appli chemic program	experi inspir pursu
program appli appli	pursu motiv pursu
program program appli	pursu passion believ
program appli program	pursu wish pursu
program appli electr	encourag pursu pursu
appli mechan program	hope continu pursu
mechan program appli	encourag pursu believ
appli electr program	passion inspir pursu
program appli chemic	desir pursu educ

The statistics presented in the next section are based on the number of applicants who mention a given topic, and hence indicate the same underlying reason for their interest in engineering

4. RESULTS

We now describe our results, treating applicants to BEE, SE and OTHER separately, as mentioned in Section 3.1. Section 4.1 presents syntactic (word frequencies and logistic regression) and semantic (question answering & clustering) results for "Engineering Interests and Goals". Section 4.5 describes the detailed analyses of programming experience (only for applicants to SE). The remaining sections discuss the results of frequency analysis and logistic regression for the remaining fields: job titles, reading interests, extracurricular activities, and additional information.

4.1 Engineering Interests and Goals

4.1.1 Syntactic Analysis

Document Frequencies: Overall, there are more terms that are used predominantly by women, indicating that women use a wider variety of terms. We see more women using non technical terms to express themselves, and men using more technical terms.

In BEE, more men mention "mechanical" (11.5% of men vs. 8.2% of women), and "compute" (8.5% of men vs. 5.3% of women. More women mention "health" (16.4% of women vs. 10.6% of men), "improve" (23.6% of women vs. 18% of men), "love" (24.8% of women vs. 20.5% of men), and "research" (20.6% of women vs. 16.5% of men).

In SE, more men mention "system" (14.2% of men vs. 9.6% of women), "problem" (25.5% of men vs. 20.9% of women), "game" (19.1% of men vs. 14.9% of women), and "goal" (25.7% of men vs. 21.5% of women). More women mention "science" (49.9% of women vs. 43.0% of men), "research" (11.0% of women vs. 6.9% of men), "challenge" (18.6% of women vs. 14.7% of men), and "people" (20% of women vs. 16.3% of men).

In the OTHER group of engineering programs, more men mention "mechanical" (28.9% of men vs. 7.5% of women), "compute" (25.8% of men vs. 17.2% of women), "robot" (16.2% of men vs. 9.9% of women),"car" (9.9% of men vs. 4.1% of women), and "goal" (24.3% of men vs. 19.4% of women). More women mention "chemical" (21.9% of women vs. 10.7% of men), "science" (41% of women vs. 32.6% of men), "creative" (16.1% of women vs. 10.2% of men), "study" (30.7% of women vs. 25.3% of men), and "love" (24.2% of women vs. 19.4% of men).

Table 9: The final set of ten topics, with representative word trigrams of the clusters classified under each topic

Reason	Trigrams (stemmed)
Family	follow footstep father older brother mechan
Contribution to Society	improv peopl live make world better make contribut societi
Outreach	attend open hous talk student professor
Technical Interests	creat new technolog water treatment plant use dismantl toy develop medic technolog
Love of Science	math physic chemistri love math scienc
Extracurriculars	book watch video robot competit team particip extracurricular activ
Prior Accomplishments	leadership communic skill profici skill mathemat
High School	talk physic teacher high school student
Professional Development	pursu graduat studi job opportun engin futur career goal
Childhood Dream	began young age dream childhood dream

Table 10: F-Measure/Accuracies for predicting gender using Engineering Interests & Goals (in %)

Group	Unigram	Bigram	Trigram
BEE	60/60.7	60/59.1	57/58
OTHER	72/78.8	76/80.4	80/77.3
SE	88/86	98/97.2	94/94

Logistic Regression: Table 10 shows the results for predicting gender using words from responses to "Engineering Interests and Goals". The predictive power of logistic regression decreases with increasing gender balance within a group, even after oversampling to compensate for the initial gender imbalance. In other words, in programs with an even gender split, it is more difficult to guess the gender.

4.1.2 Semantic Analysis

We classified the sentences extracted from students' responses under one of ten topics shown in Table 9. Table 11 shows the percentage of applicants to BEE who mentioned each topic. The most common topics are Technical Interests and Love of Science. More women mention **Love of Science**, which is statistically significant with a P-value of 0.03. No other topic had a statistically significant gender difference. On average, female students in this group

Table 11: BEE applicants' topics

Topic	%All	%Women	%Men	P-value
Family	10.9%	11.3%	10.5%	0.47
Contribution to Society	20.7%	20.5%	20.9%	0.77
Outreach	8.5%	9.3%	7.7%	0.12
Technical Interests	86.8%	86.8%	86.8%	0.97
Love of Science	**32.3%**	**34.1%**	**30.4%**	**0.03**
Extracurriculars	5.7%	5.6%	5.9%	0.69
Prior Accomplishments	6.1%	5.9%	6.3%	0.61
High School	8.8%	8.4%	9.2%	0.46
Professional Development	25.3%	25.9%	24.7%	0.47
Childhood Dream	2.5%	2.7%	2.2%	0.32

Table 12: SE applicants' topics

Topic	%All	%Women	%Men	P-value
Family	**7.6%**	**11.0%**	**7.0%**	**0.00**
Contribution to Society	12.1%	12.5%	12.0%	0.77
Outreach	8.7%	9.1%	8.6%	0.703
Technical Interests	92.6%	92.3%	92.6%	0.77
Love of Science	13.9%	16.6%	13.4%	0.05
Extracurriculars	9.2%	10.9%	9.0%	0.17
Prior Accomplishments	6.1%	7.1%	5.9%	0.30
High School	11.0%	12.9%	10.7%	0.15
Professional Development	25.0%	27.7%	24.6%	0.13
Childhood Dream	2.7%	2.8%	2.6%	0.87

Table 13: OTHER applicants' topics

Topic	%All	%Women	%Men	P-value
Family	12.3%	13.0%	12.1%	0.064
Contribution to Society	**14.7%**	**16.1%**	**14.3%**	**0.00**
Outreach	**8.1%**	**9.9%**	**7.6%**	**0.00**
Technical Interests	88.4%	89.0%	88.3%	0.149
Love of Science	**22.7%**	**26.6%**	**21.7%**	**0.00**
Extracurriculars	**9.0%**	**7.8%**	**9.3%**	**0.00**
Prior Accomplishments	6.6%	7.0%	6.5%	0.18
High School	10.3%	9.8%	10.5%	0.13
Professional Development	26.6%	27.5%	26.3%	0.07
Childhood Dream	**3.7%**	**3.0%**	**3.9%**	**0.00**

Table 14: Female students' topics across all groups

Topic	% SE	% BEE	% OTHER
Family	11.1%	11.3%	13.0%
Contribution to Society	**12.5%**	**20.5%**	**16.1%**
Outreach	9.1%	9.3%	9.9%
Technical Interests	**92.3%**	**86.8%**	**89.0%**
Love of Science	**16.6%**	**34.1%**	**26.6%**
Extracurriculars	**10.9%**	**5.6%**	**7.8%**
Prior Accomplishments	7.1%	5.9%	7.0%
High School	**12.9%**	8.4%	9.8%
Professional Development	27.7%	25.9%	27.5%
Childhood Dream	2.8%	2.7%	3.0%

mention 2.12 topics whereas male students mention 2.05, a statistically insignificant difference with a P-value of 0.06.

Table 12 shows the percentage of applicants to SE who mentioned each reason. The most common reasons are Technical Interests and Professional Development. Women mention **Family** more frequently than men, which is statistically significant with a P-value of 0.00. No other reason had a statistically significant gender difference. On average, female students in this program mention 2.04 reasons whereas male students mention 1.87, a statistically significant difference with a P-value of 0.00.

Table 13 shows the percentage of applicants to OTHER engineering programs who mentioned each topic. The most common topics are Technical Interests and Professional Development. Female students mention **Contribution to Society**, **Outreach**, and **Love of Science** more than male students, which is statistically significant with a P-value of 0.00. Male students mention **Extracurriculars** and **Childhood Dream** more than female students, which is statistically significant with a P-value of 0.00. No other topic had a statistically significant gender difference. On average, female students in this group mention 2.1 topics whereas male students mention 2.0 reasons, a statistically significant difference with a P-value of 0.00.

Table 14 highlights the differences between women who applied to BEE vs. women who applied to SE vs. women who applied to OTHER programs. The bold values show percentage differences from the other two groups that are statistically significant with a P-value of less than 0.05. Female applicants to SE, BEE, and OTHER programs differ from each other in their mentions of **Contribution to Society, Technical Interests, Love of Science, and Extracurriculars** with a P-value of less than 0.05. Mentions of **High School** are only different in SE applicants compared to other groups, which is statistically significant with a P-value of less than 0.05. No other topic had a statistically significant difference.

Table 15 highlights the differences between men who applied to BEE vs. men who applied to SE vs. men who applied to OTHER. The bold values show percentage differences from the other two groups that are statistically significant with a P-value of less than 0.05. Male applicants to SE, BEE, and OTHER programs differ from each other in their mentions of **Contribution to Society** and **Love of Science** with a P-value of less than 0.05. Mentions of **Family** and **Technical Interests** are only different for SE applicants compared to applicants to other programs, which is statistically significant with a P-value of less than 0.05. Mentions of **Extracurriculars** are different for BEE applicants compared to applicants to other program groups, which is statistically significant with a P-value of less than 0.05. No other topic had a statistically significant difference.

4.2 Reading Interests

Document Frequencies: Overall, men tend to report reading technical content such as research papers and women report reading novels and writing that has a societal focus. Words that are predominantly used by men include "article" (17.6% of men vs. 13.4% of women), "enjoy" (29.5% of men vs. 25.6% of women), "compute" (5.6% of men vs. 2.2% of women), and "science" (12.3% of men vs. 10.3% of women). Words that are predominantly used by women include "love" (20.3% of women vs. 12.6% of men), "novel"

Table 15: Male students' topics across all groups

Topic	% SE	% BEE	% OTHER
Family	**7.0%**	10.5%	12.1%
Contribution to Society	**12.0%**	**20.9%**	**14.3%**
Outreach	8.6%	7.7%	7.6%
Technical Interests	**92.6%**	86.8%	88.3%
Love of Science	**13.4%**	**30.4%**	**21.7%**
Extracurriculars	9.0%	**5.9%**	9.3%
Prior Accomplishments	5.9%	6.3%	6.5%
High School	10.7%	9.2%	10.5%
Professional Development	24.6%	24.7%	26.3%
Childhood Dream	2.7%	2.2%	**3.9%**

Table 16: F-Measures/Accuracies for predicting gender using words from Reading Interests (in %)

Group	Unigram	Bigram	Trigram
BEE	64/63	62/60	60/54.2
OTHER	79/77.8	93/89.8	95/91.8
SE	92/88.9	96/95.6	93/91.8

(31.2% of women vs. 24.6% of men), "character" (20.3% of women vs. 15.2% of men), "women" (6.1% of women vs. 1.1% of men), "people" (29.1% of women vs. 24.9% of men), and "family" (10.7% of women vs. 6.8% of men).

Logistic Regression: The results for predicting gender based on Reading Interests are shown in Table 16. As before, the predictive power of logistic regression decreases with increasing gender balance within the group.

4.3 Extracurricular Activities

Document Frequencies: Overall, male applicants' extracurricular activities have a technical focus, and female applicants have a wide breadth of experiences ranging from leadership to artistic pursuits.

In BEE, more men mention "robot" (7% of men vs. 3.6% of women) and "coach" (7.1% of men vs. 4.8% of women). More women mention "dance" (8.7% of women vs. 1.7% of men), "art" (11.3% of women vs. 6.9% of men), "council" (21.5% of women vs. 15.6% of men), and "lead" (21.1% of women vs. 16.8% of men).

In SE, more men mention "compute" (20.9% of men vs. 13.7% of women). More women mention "art" (14.5% of women vs. 4.8% of men), "council" (20.5% of women vs. 11.9% of men), "dance" (8.3% of women vs. 2.2% of men), and "lead" (18.7% of women vs. 14.3% of men).

In the OTHER group of engineering programs, more men mention "robot" (11.1% of men vs. 6.3% of women), "compute" (5.8% of men vs. 2.4% of women). More women mention "dance" (10.7% of women vs. 2.1% of men), "council" (20% of women vs. 12.1% of men), "art" (11.9% of women vs. 4.8% of men), "volunteer" (22.9% of women vs. 16.3% of men), and "lead" (19% of women vs. 13.1% of men).

Logistic Regression: The results for predicting gender based on Extracurricular Activities are shown in Table 17.

Table 17: F-Measures/Accuracies for predicting gender using words from Extracurricular Activities (in %)

Group	Unigram	Bigram	Trigram
BEE	72/72.9	69/66.6	62/59.5
OTHER	81/81.1	80/77.8	78/71.4
SE	85/83.3	85/82	94/93.4

Table 18: F-Measures/Accuracies for predicting gender using words from Job Titles (in %)

Group	Unigram	Bigram	Trigram
BEE	59/57.9	58/52.6	63/51
OTHER	65/61.5	64/59.1	67/61.9
SE	67/63.7	66/58.7	68/51.7

The predictive power of logistic regression decreases with increasing gender balance within the group.

4.4 Job Titles

Document Frequencies: Across all programs, men are more likely to mention terms that imply technical work or manual labour, whereas women are more likely to mention terms that imply customer service and caring professions. Example words in job titles from male applicants include "referee" (4.1% of men vs. 2% of women), "labor" (2.6% of men vs. 0.5% of women), and "technician" (3.1% of men vs. 1.2% of women). Example words in job titles from female applicants include "cashier" (12.8% of women vs. 6.8% of men), "teacher" (6.2% of women vs. 2.7% of men), and "assist" (17.6% of women vs. 14.3% of men).

Logistic Regression: As shown by the logistic regression scores in Table 18, Job Titles do not provide as much predictive power as other fields.

4.5 Programming Experience

4.5.1 Syntactic Analysis

Document Frequencies: In general, women use more non technical terms, and men use more technical terms. Examples of terms that are more commonly used by male applicants include "game" (30.8% of men vs. 22.3% of women) and "develop" (21.5% of men vs. 14.4% of women), and terms more commonly used by female applicants include "mark" (39.9% of women vs. 30.6% of men) and "attend" (4.2% of women vs. 1.4% of men). Through manual inspection, we discovered that "mark" was used in the context of earning a certain mark in a course. "attend" was used to indicate attendance in a programming workshop or event.

Logistic Regression: As shown in Table 19, the words used to describe programming experience can be used to predict the gender of the applicant.

4.5.2 Programming Languages

Table 20 shows a comparison of specific language knowledge between male and female applicants. All languages except for SQL are slightly skewed toward male applicants; however, only **Java, C++, C, Turing, C#** have statistically

Table 19: F-Measures/Accuracies for predicting SE applicants' gender using Programming Experience (in %)

Group	Unigram	Bigram	Trigram
SE	91/88.8	98/98	97/95.7

Table 20: Comparison of reported programming language knowledge

Language	% Women	% Men	Difference	P-value
Java	**58.9%**	**65.6%**	**-6.7%**	**0.00**
C++	**23.3%**	**28.5%**	**-5.2%**	**0.01**
Python	25.1%	28.1%	-3.0%	0.18
HTML/CSS	19.0%	19.5%	-0.5%	0.75
Basic	16.1%	18.6%	-2.5%	0.114
C	**12.5%**	**17.0%**	**-4.5%**	**0.01**
JavaScript	12.7%	15.0%	-2.3%	0.17
Turing	**10.8%**	**14.3%**	**-3.5%**	**0.03**
C#	**6.1%**	**9.4%**	**-3.3%**	**0.01**
Php	**3.9%**	**8.3%**	**-4.4%**	**0.00**
SQL	3.9%	3.2%	-0.7%	0.50
Other	31.1%	16.4%	-3.3%	0.07
None	4.7%	3.2%	+1.4%	0.07

significant differences with a P-value of less than 0.05. In these cases, we only see differences ranging from 4% to 6%.

Men on average report experience with 2.43 programming languages, whereas women report experience with 2.05 languages, a significant result with a P-value of less than 0.05.

4.5.3 Programming Concepts

Among applicants who mentioned specific programming concepts, women reported **Basic Language Knowledge**, which includes loops, if-statements, and variables, 14% more than male applicants did. This result is significant with a P-value of less than 0.05.

There are small differences in mentions of data science, object oriented programming, sorting, searching, and data structures. However, these results were not statistically significant, so we cannot conclude that there is a gender difference in any mention of programming concepts.

4.5.4 Learning Method

We found that men were slightly more likely to learn how to program through employment or self-learning, and women were more likely to learn how to program in high school, through higher education, and through online courses. This result is not statistically significant with a P-value of greater than 0.05, so we cannot conclude that there is a gender difference in how men and women learn how to program.

4.5.5 Experience

On average, women report 6 months of programming experience, and men report 8 months of programming experience. This result is not significant with a P-value of greater than 0.05, so we cannot conclude that there is a gender difference in the amount of experience within applicants to SE.

Table 21: F-Measures/Accuracies for predicting gender using Additional Information (in %)

Group	Unigram	Bigram	Trigram
BEE	60/58.4	52/51.3	53/50
OTHER	78/77.3	81/77	93/89.2
SE	86/83.7	93/86.3	93/95.2

4.6 Additional Information

Document Frequencies: We see a difference in word choice between men and women when answering a question with no restrictions on the content of their answer.

In BEE, more men mention "sport" (10.9% of men vs. 7.1% of women) and "compute" 4.7% of men vs. 2.3% of women). More women mention "educate" (17.2% of women vs. 12.2% of men), "science" (17.9% of women vs. 13.4% of men), "develop" (15.1% of women vs. 10.7% of men), "community" (14.8% of women vs. 10.8% of men), and "create" (8.5% of women vs. 5.0% of men).

In SE, more men mention "compute" (27.8% of men vs. 20.8% of women) and "game" (9.2% of men vs. 3.8% of women). More women mention "attend" (16.7% of women vs. 10.4% of men), "English" (12.8% of women vs. 7.2% of men), "study" (21.5% of women vs. 16.5% of men), "parent" (8.7% of women vs. 3.7% of men), "love" (14.2% of women vs. 10.1% of men), and "creative" (8.7% of women vs. 4.6% of men).

In the OTHER programs, more men mention "sport" (10.2% of men vs. 5.7% of women) and "team" (16.3% of men vs. 12.4% of women). More women mention "art" (7.3% of women vs. 3.3% of men), "volunteer" (9.9% of women vs. 6.4% of men), and "passion" (13.6% of women vs. 10.4% of men).

Logistic Regression: The results for predicting gender based on Additional Information are shown in Table 21. As before, the predictive power of logistic regression decreases with increasing gender balance within the group.

5. DISCUSSION

5.1 Similarities

Regardless of gender, the most commonly mentioned topic in responses to "Why are you interested in engineering?" is Technical Interests. Female and male applicants seem to share the same interest in Engineering in all program groups. SE applicants show more technical interest in engineering than other programs.

In general, female and male applicants to SE mention the same motivation for studying engineering. Family is more popular among female applicants, not because female applicants to SE mention it more compared to other programs, but because male applicants talk about it less than men in other programs, as can be seen in Tables 15 and 14.

In SE, we do not see a large gender gap in self reported programming experience, as shown in Table 20. This suggests that students who are exposed to computer science do not differentiate themselves through the number of languages they learn, nor in the amount of programming experience.

In BEE, the differences between female and male applicants are minimal. We see evidence for this in the semantic analysis presented in Section 4.1.2 where there is only one topic that shows a gender difference, and we observe this in our inability to reliably predict gender based on any question as shown in Tables 11, 18, 17, and 21.

Based on Tables 14 and 15, Contribution to Society and Engineering Interests are inversely proportional, regardless of gender.

5.2 Differences

5.2.1 Depth vs. Breadth

The overarching gender difference throughout the analysis is that men differentiate themselves through depth of experience, and women through breadth of experience. To study engineering, all applicants must demonstrate knowledge in mathematics and sciences through their academic work. However, we see male applicants differentiating themselves by highlighting their initiative to acquire more technical skills through their work experience, extracurricular activities, reading interests, and the topics they mention when asked why they are studying engineering. Female applicants differentiate themselves through demonstrating a wide range of experiences and capabilities. This is suggested by the fact that women mention a wider variety of topics when asked why they are studying engineering, their extracurricular activities place an emphasis on leadership and artistic pursuits, they often take service jobs, and they choose to discuss more non technical reading material.

In SE, men are more likely to report technical extracurriculars, as seen in Section 4.3, even though there is only a small gender difference in the reported amount of programming experience. This provides further justification that women differentiate themselves through breadth of experience even when they are extremely technically focused.

The gender difference in depth versus breadth is much smaller in BEE. The difference in the number of topics mentioned between men and women is the smallest across these two programs. We also only see a statistically significant difference between men and women in one topic, love of science, which is extremely common across all applicants. The small difference is consistent with our inability to predict gender in BEE.

We also see this in the syntactic analysis of reasons, where women mention “improve” and “health” more in the BEE group, and “people” more in the SE group. It is an interesting difference because BEE includes programs that focus on helping others, and SE is often the farthest removed from directly working with people.

5.2.2 Desire to Serve Society

Women show a stronger desire to contribute to society and improve the world around them. We see this in their motivation to study engineering in “Engineering Interests and Goals” in the OTHER group of programs where they are more likely to mention “Contribution to society”. We also see this in the syntactic analysis of this field where they mention “improve” and “health” in the BEE group, and “people” in the SE group. This is also evident in their work experience where women mention “assist” and “teacher” more often than men. Finally, we see this in extracurricular activities, where women mention “volunteer” more frequently than men. Our findings in this section agree with [3, 4].

5.2.3 Influence

Women are more likely to mention personal influences in their decision to study engineering. This is prevalent in SE, where women mention “Family” reasons more than men. This expands on the findings in [19].

6. CONCLUDING REMARKS

The main findings of this paper are that men differentiate themselves through having technical depth in their experiences, and women differentiate themselves through having a breadth of experiences. We see similar behavior in Software Engineering, even though women and men show similar levels of technical know-how. We see smaller gender differences in applicants to Biomedical and Environmental Engineering where there is gender equity. Finally, women mention more of a desire to serve society, and they mention more interpersonal reasons for studying engineering than men.

We infer that in order to attract more women to study engineering, it must be presented as a profession that can help others and allow for a broad range of careers and learning opportunities. A key part in fostering this new image of engineering lies in encouragement from family and role models who practice engineering.

6.1 Future Work

In future work, we intend to conduct data driven analysis of gender differences at various stages in STEM students' academic careers; e.g., investigating the effects of university-sponsored outreach and mentorship programs on applicants, and correlating depth and breadth of expression at the time of admission to academic and career success. We also plan to investigate and compare gender differences in graduate school applications to those in undergraduate applications. We also want to expand the scope of our studies to include non STEM programs in our analysis, and conduct comparisons of differences in STEM vs. non-STEM programs.

7. REFERENCES

[1] T. Busch. Gender differences in self-efficacy and attitudes toward computers. *Journal of Educational Computing Research*, 12(2):147–158, 1995.

[2] N. Chinchor. Muc-4 evaluation metrics. In *Proc. of the 4th Conf. on Message Understanding*, 1992.

[3] A. B. Diekman, E. R. Brown, A. M. Johnston, and E. K. Clark. Seeking congruity between goals and roles: A new look at why women opt out of science, technology, engineering, and mathematics careers. *Psychological Science*, 21(8):1051–1057, 2010.

[4] J. Eccles. Where are all the women? gender differences in participation in physical science and engineering. In *Why aren't more women in science?:*

Top researchers debate the evidence, pages 199–210. American Psychological Association, 2007.

[5] M. Feng, J. Roschelle, C. Mason, and R. Bhanot. Investigating gender differences on homework in middle school mathematics. In *Proc. of the Int. Conf. on Educational Data Mining (EDM)*, pages 364–369, 2016.

[6] J. L. Fleiss, B. Levin, and M. C. Paik. Determining sample sizes needed to detect a difference between two proportions. In *Statistical Methods for Rates and Proportions*, pages 64–85. John Wiley & Sons, Inc., 2004.

[7] P. W. Foltz, W. Kintsch, and T. K. Landauer. The measurement of textual coherence with latent semantic analysis. *Discourse Processes*, 25(2-3):285–307, 1998.

[8] A. Genkin, D. D. Lewis, and D. Madigan. Large-scale bayesian logistic regression for text categorization. *Technometrics*, 49(3):291–304, 2007.

[9] D. Hango. Gender differences in science, technology, engineering, mathematics, and computer science (STEM) programs at university. *Insights on Canadian Society*, 12 2013.

[10] S. Hussain, J. Hazarika, and P. Buragohain. Educational data mining on performance of under graduate students of dibrugarh university using r. *International Journal of Computer Applications*, 114(11):10–16, 2015.

[11] T. K. Landauer and S. T. Dumais. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240, 04 1997.

[12] Z. J. Kovacic. Early prediction of student success: Mining students enrollment data. In *Proc. of Informing Science & IT Education Conference (InSITE)*, 2010.

[13] H. M. Matusovich, R. A. Streveler, and R. L. Miller. Why do students choose engineering? a qualitative, longitudinal investigation of students' motivational values. *Journal of Engineering Education*, 99(4):289–303, 2010.

[14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[15] J. Ramos. Using TF-IDF to determine word relevance in document queries. In *Proc. of the Instructional Conf. on Machine Learning*, volume 242, pages 133–142, 2003.

[16] M. Saarela and T. Karkkainen. Discovering gender-specific knowledge from finnish basic education using PISA scale indices. In *Proc. of the Int. Conf. on Educational Data Mining (EDM)*, pages 60–67, 2014.

[17] M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603, 2016.

[18] L. Shashaani. Gender differences in computer attitudes and use among college students. *Journal of Educational Computing Research*, 16(1):37–51, 1997.

[19] A. Y. Smith. *They chose to major in engineering: A study of why women enter and persist in undergraduate engineering programs*. PhD thesis, 2012.

[20] A. Sullivan and M. U. Bers. Girls, boys, and bots: Gender differences in young children's performance on robotics and programming tasks. *Journal of Information Technology Education: Innovations in Practice*, 15:145–165, 2016.

[21] B. Trstenjak, S. Mikac, and D. Donko. KNN with TF-IDF based framework for text categorization. *Procedia Engineering*, 69:1356–1364, 2014.

[22] T. Zimmermann. Card-sorting: From text to themes. In *Perspectives on Data Science for Software Engineering*, pages 137–141. Elsevier Science, 2016.

A Comparison of Features for the Automatic Labeling of Student Answers to Open-ended Questions

Jesus Gerardo Alvarado Mantecon
University of Ottawa
800 King Edward Avenue.
Ottawa, Canada.
+1 613 668 7214
jalva061@uottawa.ca

Hadi Abdi Ghavidel
University of Ottawa
800 King Edward Avenue.
Ottawa, Canada.
+1 613 890 1389
habdi018@uottawa.ca

Amal Zouaq
University of Ottawa
800 King Edward Avenue.
Ottawa, Canada.
+1 613 562 5800 ext.6227
azouaq@uottawa.ca

Jelena Jovanovic
University of Belgrade
Jove Ilica 154.
11000 Belgrade, Serbia.
+381 11 3950 853
jelena.jovanovic@fon.bg.ac.rs

Jenny McDonald
University of Auckland
Private Bag 92019.
Auckland, NZ.
+64 9 9238140
j.mcdonald@auckland.ac.nz

ABSTRACT

The automatic evaluation of text-based assessment items, such as short answers or essays, is an open and important research challenge. In this paper, we compare several features for the classification of short open-ended responses to questions related to a large first-year health sciences course. These features include a) traditional n-gram models; b) entity URIs (Uniform Resource Identifier) and c) entity mentions extracted using a semantic annotation API; d) entity mention embeddings based on GloVe, and e) entity URI embeddings extracted from Wikipedia. These features are used in combination with classification algorithms to discriminate correct answers from incorrect ones. Our results show that, on average, n-gram features performed the best in terms of precision and entity mentions in terms of f1-score. Similarly, in terms of accuracy, entity mentions and n-gram features performed the best. Finally, features based on dense vector representations such as entity embeddings and mention embeddings obtained the best f1-score for predicting correct answers.

Keywords

Short open-ended responses, N-gram models, Entity URIs, Entity Mentions, Entity embeddings, Mention embeddings.

1. INTRODUCTION

Due to the growth of Massive Open Online Courses (MOOCs) and increased class sizes in traditional higher education settings, the automatic evaluation of answers to open-ended questions has become an important challenge and one which has yet to be fully resolved. On the other hand, it has been shown that open-ended assessments are better able to capture a higher level of understanding of a subject than other machine-scored assessment items [24]. Still, MOOCs usually rely on multiple-choice questions since the evaluation of open-ended assessments requires more resources in massive online courses [32]. The human effort required to manually evaluate students' answers has escalated with the spread of large-scale courses that enroll several hundred, or even thousands of students. To tackle this challenge, we analyze textual responses to a set of open-ended questions designed to encourage deep responses from students. We explore the use of vector space models (VSMs) that represent each answer with a real-valued vector, and evaluate those models on the task of classifying student responses into correct and not-correct. In particular, we examine and evaluate different feature sets that can be automatically derived from students' answers and used to represent those answers as vectors in a high dimensional space. The examined features do not require handcrafting based on the particularities of specific questions. Our main objective is to examine and compare the predictive power of different text features, automatically extracted from a corpus of answers to open-ended questions, on multiple classification algorithms.

We build VSMs using different text representations that result in either a sparse VSM (e.g., n-gram based VSM) or a dense VSM (e.g., VSM based on word embeddings). For sparse VSMs, we explore traditional n-gram features (unigrams, bigrams, trigrams, and n-grams that combine all of the previous features). We also investigate the usefulness of semantic annotations of students' responses for the classification task. Semantic annotation adds machine-readable meaning in the form of entities [21]. Hence, it enables the association of students' answers with vectors of domain-specific entities. Semantic annotators often rely on open Web-based knowledge bases such as DBpedia [13], an RDF representation of Wikipedia's semi-structured content. For example, given the entity *Aorta,* identified by a semantic annotator, we obtain its associated Web resource from DBpedia (http://dbpedia.org/page/Aorta), which further links to other related entities and properties from DBpedia. We make use of two semantic annotators: DBpedia Spotlight [19] and TAGME [6]. We query each annotator with the students' responses to obtain entities mentioned in the response. For each entity, we take the entity label and use it as *entity mention feature*, whereas the entity's Uniform Resource Identifier (URI) is used as *entity URI* feature.

To build a dense VSM, we rely on the entity mentions identified through semantic annotation and pre-trained word and entity embeddings. In particular, we retrieve vector representations of entity mentions using a GloVe model pre-trained on Wikipedia dumps [23]. Thus, our fourth feature set consists of entity mention embeddings based on GloVe. Finally, we represent entity URIs using a Wiki2Vec model trained on Wikipedia dumps to obtain another dense VSM. Hence, entity URI embeddings extracted from Wikipedia constitute our fifth feature set.

Given the short length of most answers and large vocabularies providing sparse vectors, we decided to include the last two sets of features to produce dense vector representations. In fact, dense vectors have shown an increase in performance for several natural language processing tasks [15]. Both GloVe [23] and Word2vec models [20] learn vector representations of words (called word embeddings) based on context. In total, we compare five types of features (n-gram, entity mentions, entity URIs, mention embeddings and entity embeddings) to train classification models to automatically label each student answer as correct or incorrect.

The rest of the paper is structured as follows: In Section 2, we present related work on automatic short answer grading. Then, we introduce our methodology, including the corpus description, our analysis pipeline and an in-depth description of our features. Section 5 describes the results of our experiments followed by the analysis of the effect of feature selection on our classifiers in Section 6. Finally, we discuss our findings and conclude the paper in Section 7 and 8.

2. RELATED WORK

One of the hot topics in the field of educational data mining is automatic short answer (response) grading (ASAG). In general, there are two kinds of ASAG approaches: response-based and reference-based [27]. In this paper, we analyze students' answers based on the response-based approach, which focuses only on students' answers. In contrast, reference-based ASAG also rely on the comparison of the student answer to the model answer.

Burrows et al. [4] classified all types of approaches to ASAG into five categories (eras): Concept mapping [8, 10, 12], Information extraction [5], Corpus-based methods [11], Machine learning, and Evaluation [28, 30]. In the Machine Learning approach, which is the approach followed in this study, the trend is to build models (supervised or unsupervised) through data mining and natural language processing techniques in order to assess students' answers.

ASAG systems can also be categorized into semi-automatic (teacher-assisted) and fully-automatic systems. In semi-automatic systems, students' answers are processed (clustered) to facilitate the grading process. For example, Basu [1] applied k-medoids clustering to students' answers to ease the grading process. In another work, Jayashankar [9] proposed an integration of data mining and word clouds to help teachers evaluate student answers through visualization.

Fully-automatic systems produce grades for each student, with or without additional feedback. Several features are considered in training these systems: lexical features (e.g. word length), syntactic features (e.g. sentence length and part-of-speech), semantic features (e.g. semantic annotations and triples), discursive features (e.g. referential expressions), statistical features (e.g. language modelling like n-grams and embeddings), and similarity features (e.g. cosine similarity).

McDonald et al. [17, 18] evaluated Naive Bayes and Max Ent classifiers using a number of features like bag of words, word length, and word and character n-grams. Madnani et al. [14] used these types of features in combination with triples to examine the performance (accuracy) of 8 different classifiers and regressors (linear and nonlinear). In another work, Riordan et al. [26] combined n-gram features, answer length, and word and character embeddings to compare the performance of SVM (as a baseline) with neural architectures. In several approaches, features based on the similarity between the students' responses and the teacher's response were used together with n-grams. For example, Sakaguchi et al. [27] used stacked generalization [31] to integrate response-based and reference-based models. In particular, Sakaguchi et al. first built a classifier based on sparse response-based features (e.g. character n-gram and word n-gram); the obtained predictions were combined with dense reference-based features (e.g. BLEU [22]) to build another stacked classifier. Both classifiers were built as support vector regression (SVR) models. Zhang et al. [33] compared Deep Belief Networks (DBN) [2] to five classifiers such as Naive Bayes and Logistic Regression. The classifiers were trained on features extracted from three models, namely the Question model (e.g. question difficulty), the Student model (e.g. probability that a student learned a concept based on the student's past performance), and the Answer model (e.g. length difference between student answer and model answer). The DBN performed better than the other classifiers in terms of accuracy, precision, and F-measure, but not recall. Roy et al. [25] developed an ASAG system that can grade answers in different domains. They relied on an ensemble classifier of student answers (question-specific approach) and a numeric classifier based on the similarity score between the model answer and students' answers (question-agnostic approach). Their features were words, n-grams, and similarity scores between student answers and model answer. Finally, Tack et al. [29] used ensemble learning of five classifiers based on lexical features (e.g., word length), syntactic features (e.g., sentence length), discursive features (e.g., number of referential expressions), and a number of psycholinguistic norms.

In this work, we follow the response-based approach as we build classifiers based on students' answers. Our approach differs from previous works in that we carry out ASAG (and more specifically classification) by comparing six classifiers trained with both sparse vector representations (based on n-grams and entities) and dense vectors representations (GloVe, Word2Vec). One additional difference is the use of semantic annotations (entity mentions and entity URIs) to build some of our vector space models. Finally, the features used in this work do not necessitate a huge feature engineering effort as they come directly from text or from the use of a semantic annotation API and an embedding model.

3. METHODOLOGY

We first give a description of the corpus used in our experiments, then we detail our overall approach as well as the metrics used in the evaluation phase. This is followed by an in-depth explanation of our features.

3.1 Corpus Description

Our data set is extracted from a corpus of student short-answer question (SAQ) responses drawn from a first-year human biology course (McDonald [16]). Among multiple elements in our data set, our experiments are based only on the labeled student responses to the survey and model answers (expected answers to the questions). Student SAQ responses and associated metadata were collected through a dialog system.

From the initial data set, we selected a sub-set of student answers based on the following criteria:

- Answers from the year 2012 only as this year is the one with the highest participation; out of 15,758 answers collected over 4 years, 7,548 originate from 2012.
- Out of the 42 different unique questions, we only use 6 questions that provide a reasonable number of responses as well as lengthy (deep) responses. We avoided questions that do not encourage answers that display deep understanding of the topic (e.g., yes-no questions, calculation questions or multiple choice questions).

The questions asked are designed to encourage deep responses from students [3]. The students are expected to explain or describe the knowledge obtained during the course in their own words rather than giving answers by the book. Table 1 presents the questions used in the study and their expected answers.

Table 1. Survey questions

ID	Question	Model Answer
Q.1	HR or heart rate is the number of times the heart beats each minute. A normal adult HR is around 72 beats/min. How would you check someone's HR?	You could measure their pulse.
Q.2	What is the pulse?	The pulse is a pressure wave or a pulsatile wave generated by the difference between systolic and diastolic pressures in the aorta.
Q.3	Inotropic state is a term that is sometimes used to describe the contractility of the heart. Can you describe what is meant by contractility?	Contractility is the force or pressure generated by the heart muscle during contraction.
Q.4	If you were 'building' a human being and you wanted to position receptors in the body to monitor blood pressure, where would you put them?	You'd probably want to put them near vital organs and at the main outflow from the heart. It turns out that the main human baroreceptors are located in the carotid sinuses and aortic arch.
Q.5	What feature of artery walls allows us to feel the pulse?	Artery walls are thick and strong and not very compliant
Q.6	Can you explain why you cannot feel a pulse in someone's vein?	You cannot feel a pulse in veins because the blood flow in veins is not pulsatile

The resulting sub-set amounts to 1,876 answers from 218 students to 6 questions. Note that not all students answered all the questions. Completing responses was voluntary, which accounts for the variability in the number of responses received to each question. In addition, the nature and quality of the responses are not necessarily representative of the class as a whole. Table 2 presents descriptive statistics on the students' answers to the selected subset of questions used in all the experiments.

Table 2. Statistics on students' answers per question

Question	Avg. words	Min. words	Max. words	Answers	Correct (%)
Q.1	6	1	36	243	65.43%
Q.2	9	1	82	422	17.54%
Q.3	6	1	31	316	33.86%
Q.4	4	1	34	151	54.97%
Q.5	3	1	27	171	25.15%
Q.6	9	1	34	361	31.86%

Each of these questions is associated with a set of students' answers. As an example, for question Q.6, we present the expected answer (i.e. Model answer), a deep response (Student Answer 1), and a simpler response (Student Answer 2):

Model Answer: You cannot feel a pulse in veins because the blood flow in veins is not pulsatile

Student Answer 1: The wave motion associated with the heart beat is stopped by the arteries and capillaries. Therefore, the vein has no pulse.

Student Answer 2: The blood flow is continuous.

Both student answers were labeled as *correct* by the human markers. Student Answer 1 would be considered a deeper answer than Student Answer 2, because it makes explicit the reasoning behind the answer, thus suggesting a better understanding of the topic.

The students' responses were manually evaluated by human markers with expertise in the domain of human biology. The annotators assigned a label negotiated through discussion. Such labels describe different aspects of an answer like quality of the response or correctness [16]. For example, answers may be labelled as incorrect, incomplete, and display disinterest in responding (*dont-know* label), among others. Further details on the labels used can be found in McDonald [16]. Table 3 displays some of those answers and the assigned labels.

Table 3. Student Answers sample

Question	Student Answer	Label
Q.5	Lack of elastic tissue	incorrect
Q.6	idk lol	dont-know
Q.4	In major arteries of the body, such as the common carotid or the aortic arch	ok
Q.3	ability to change volume	incomplete
Q.6	Ventricle contracts blood ejected into aorta, expanding vessel and increase pressure in vessel, wave of pressure cane felt is pulse	correct

For all of our experiments, we used model answer (expected answer) and student answers and re-labeled them as *correct* or *not-correct. Correct* answers comprise model answers plus all answers labeled as *correct* or *ok*. All other answers were re-labeled as *not-correct*. The resulting data set is composed of 65% *not-correct* answers and 35% *correct* answers.

3.2 Overall Approach

Our general approach can be described as follows:

1. ***Data pre-processing:*** in this step, we perform lemmatization and removal of punctuation marks and stop words (NLTK[1] stop words list) from the selected answers.
2. ***Feature extraction***: We consider five types of features: n-gram, entity URIs, entity mentions, URI embeddings, and mention embeddings, which are detailed in section 4. We extract n-grams, entity URIs and entity mentions from student responses. Then, entity mentions are used to query a pre-trained GloVe model [23] to obtain mention embeddings. Likewise, entity URIs are used to query a pre-trained Wiki2Vec model [34] to obtain entity embeddings. Both GloVe and Wiki2Vec are pre-trained on Wikipedia.
3. ***Vector space model (VSM)***: For n-gram features, entity mentions, and entity URIs, we compute a vector representation of each answer by extracting a vocabulary from all students' answers and using TF-IDF as the relevance metric to weight each feature in an answer. As for mention embeddings and entity embeddings, we generate VSMs by averaging embeddings over all mentions or URIs appearing in an answer. The output from this step is one VSM representation of all answers for each feature type.
4. ***Classification task***: we run several classification algorithms: the ZeroR algorithm as our baseline, Logistic regression, K-nearest neighbors (IBK), Decision trees (J48), Naïve Bayes, Support vector machine (SVM), and Random forest. We train each classifier using the entire data set of answers regardless of the question to which they belong. The rationale is that all answers belong to the same domain, and thus can be expected to be in a shared semantic space.

3.3 Evaluation Metrics

The evaluation is performed through 10-fold cross validation on each classifier. The metrics used for this purpose include:

- Accuracy: Percentage of correctly classified answers.
- Area Under the Curve (AUC): Probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance.
- Precision: Fraction of correctly classified answers within all classified instances.
- Recall: Fraction of relevant answers successfully retrieved.
- F1-score: Weighted harmonic mean of the precision and recall. It represents how precise and complete a classifier is.

4. FEATURE DESCRIPTION

4.1 N-gram Features

We create a vector representation for each answer based on n-grams. Table 4 shows some descriptive statistics on the obtained n-grams. We perform four experiments using different n-grams: unigrams, bigrams, trigrams, and the combination of all of them. To that end, four VSMs are built, one per n-gram group. Each vector holds the TF-IDF value of each item found in the answers. TF-IDF is calculated with the formula:

$$tf\text{-}idf_{i,j} = tf_{i,j} \times (\log \frac{1+n_d}{1+df_i} + 1)$$

Where $\boldsymbol{tf_{i,j}}$ is the total number of occurrences of the term i in the student answer j, $\boldsymbol{n_d}$ is the total number of documents (i.e. answers) and $\boldsymbol{df_i}$ is the number of documents (i.e. answers) containing the term i.

Table 4. Total number of n-grams in answers for all questions

Answers	Unigrams	Bigrams	Trigrams
Unique	700	2114	4750
Total	6364	2589	3383

4.2 Entity URI Features

These features are based on entity URIs extracted from answers using two semantic annotators: DBpedia Spotlight and TAGME (see Sect. 1). The basic unit in the built VSM is the URI of a DBpedia resource (e.g. http://dbpedia.org/page/Baroreceptor). We send get requests to both annotators with the answers to be analyzed, and receive, for each answer, a list of entity mentions and their associated URIs. Table 5 shows statistics on the number of entity URIs and mentions (lowercase) retrieved by each of the two annotators.

Table 5. Number of entity URIs and mentions on all answers

Semantic Annotator	Entities		Mentions	
	Unique	**Total**	**Unique**	**Total**
Spotlight	143	1620	188	1620
TAGME	876	5054	806	5054

Table 6 provides an example of retrieved entity mentions and URIs for an answer to Q.2.

Table 6. Sample of retrieved entity URIs

Answer	Semantic Annotator	Mention	URI
Recoil caused by pressure in arteries	Spotlight	Recoil	dbpedia.org/page/Recoil
		Arteries	dbpedia.org/page/Artery
	TAGME	Recoil	dbpedia.org/page/Recoil
		Pressure	dbpedia.org/page/Pressure
		Arteries	dbpedia.org/page/Artery

We build a vector representation of each answer for each of the following configurations (i.e., vocabularies):

- Spotlight_URI: Set of entity URIs retrieved from all answers using DBpedia Spotlight.
- TAGME_URI: Set of entity URIs retrieved from all answers using TAGME.
- Intersection: Set intersection between the entity URIs retrieved from all answers with both tools.
- Union: Set union between the entity URIs retrieved from all answers with both tools.

This produces four VSMs based on entity URIs. The resulting VSMs use TF-IDF as the metric for estimating the value of each entity URI for each answer.

[1] https://www.nltk.org/

4.3 Entity Mention Features

We use the annotations retrieved by Spotlight and TAGME, selecting entity mentions as the basic units for building VSMs. A mention is a sequence of words spotted in an answer and associated to a URI. This means that an entity mention can be a unigram, but also a bigram or trigram. We compute the TF-IDF of each entity mention present in an answer to build its vector representation. As in entity URI features, we have one vocabulary per configuration with four VSMs as the final output. The available configurations, based on mentions, used to build a vector representation of each answer, are analogous to those described for entity URIs, except that they are based on mentions (Spotlight_Mention, TAGME_Mention, Intersection and Union).

4.4 Entity Embedding Features

For this set of features, we rely on the Wiki2Vec[2] model, a Word2Vec implementation pre-trained on Wikipedia, where Wikipedia hyperlinks are replaced with DBpedia entities (URIs). The model was presented by Zhou et al. [34] and is based on 100-dimensional vectors. Word2Vec models can either learn to predict a word given its context (CBOW) or predict a context given a target word (Skip-gram) [20]. This creates a vector space in which similar words or entities are close to each other. Likewise, Wiki2Vec creates a vector space model in which similar DBpedia entities are close to each other. Given that our entity URIs reference DBpedia resources, we consider it a suitable match. For each configuration, we query the Wiki2Vec model with the entity URIs found in each answer to obtain their corresponding embeddings. Table 7 shows the percentage of entity URIs that are associated with an embedding vector in the Wiki2Vec model per configuration. We also show the percentage for the GloVe model which is presented in section 4.5.

Table 7. Coverage of entity URIs and mentions on their corresponding models (Wiki2Vec and GloVe)

Configuration	% of entity URIs in Wiki2Vec	% of entity mentions in GloVe
Spotlight_URI	97.5 %	50.46%
TAGME_URI	93.94 %	62.16%
Intersection	97.11 %	49%
Union	94.63 %	65.10%

For each configuration, we have one VSM. In each VSM, we aggregate the entity embeddings per answer by calculating the average of the entity URI vectors. This produces a single embedding that represents the answer.

4.5 Mention Embedding Features

For the mention embedding features, we rely on word embeddings, where each word is an entity mention instead of an entity URI. We use the GloVe model [23] trained using Wikipedia dumps from 2014 and build vectors with 100 dimensions (as for entity URI embeddings). Unlike Word2Vec, GloVe is a count-based model derived from a co-occurrence matrix. We query the GloVe model with the entity mentions found in each answer. The coverage of the model is given in Table 7. For each configuration, we have one VSM where each answer is represented as the average of the entity mention vectors.

5. RESULTS

For each feature set, we trained six classification algorithms, and evaluated 120 different models. Due to the space limit, we present only the top two performing classifiers (Random forest and SVM) in terms of overall accuracy for each of our feature sets. ZeroR is also included as the baseline.

5.1 N-gram Results

Table 8 shows the accuracy (ACC) and AUC obtained using n-gram features. Overall, the accuracy and AUC obtained with Random forest were always higher than with SVM. In particular, unigrams obtained the best accuracy of 88.40% as well as the highest AUC (.95) using Random forest.

Table 8. Accuracy & AUC using n-gram features

N-gram	Random Forest		SVM		ZeroR	
	ACC %	AUC	ACC %	AUC	ACC %	AUC
Unigrams	**88.40**	**.95**	84.44	.82	65.1	.50
Bigrams	81.97	.87	79.93	.73	65.1	.50
Trigrams	72.84	.68	72.12	.61	65.1	.50
N-grams	85.58	.93	84.25	.80	65.1	.50

Table 9 shows additional results for models cross-validated with n-gram features. For the *correct* label, our best classifier was Random forest using unigrams for the f1-score (.82) and trigrams or n-grams for best precision (.93). For the *not-correct* label, again, Random forest got the best results, using unigrams for both f1-score (.91) and precision (.88).

Table 9. Precision, recall & f1-score using n-gram features

Label	Classifier	N-gram	Precision	Recall	F1
Correct	Random Forest	Unigrams	.89	**.77**	**.82**
		Bigrams	.92	.53	.67
		Trigrams	**.93**	.24	.38
		N-grams	**.93**	.63	.75
	SVM	Unigrams	.79	**.76**	**.77**
		Bigrams	.85	.51	.64
		Trigrams	**.88**	.23	.37
		N-grams	.86	.65	.74
	ZeroR	Unigrams	0	0	0
		Bigrams	0	0	0
		Trigrams	0	0	0
		N-grams	0	0	0
Not-correct	Random Forest	Unigrams	**.88**	.95	**.91**
		Bigrams	.79	.98	.88
		Trigrams	.71	**.99**	.83
		N-grams	.83	.98	.90
	SVM	Unigrams	**.87**	.89	.88
		Bigrams	.79	.95	.86
		Trigrams	.71	**.98**	.82
		N-grams	.84	.95	**.89**
	ZeroR	Unigrams	.65	1	.79
		Bigrams	.65	1	.79
		Trigrams	.65	1	.79
		N-grams	.65	1	.79

[2] https://github.com/idio/wiki2vec

A visible drop in recall from unigrams to trigrams (difference of .53) can be spotted for the correct label in both SVM and Random Forest. Based on the number of elements in each n-gram feature (Table 4), we observe that the amount of bigrams and trigrams is notably lower than unigrams. This can, at least partially, explain the lower recall using these features. Another noticeable result is that while the results obtained with Random Forest and SVM exceed the baseline for the *correct* label in terms of precision, recall and f1-score, the results for the *not-correct* label are closer to the baseline.

5.2 Entity Mention Results

The highest accuracy among these feature sets was achieved by Random Forest with the Union configuration (88.58%), as shown on Table 10. Again, Random forest outperformed SVM in terms of accuracy and AUC for each configuration.

Table 10. Accuracy & AUC using Entity mentions

Tool	Random Forest		SVM		ZeroR	
	ACC %	AUC	ACC %	AUC	ACC %	AUC
Spotlight Mention	78.61	.78	75.05	.67	65.1	.50
TAGME Mention	88.52	**.95**	85.22	.83	65.1	.50
Intersection	78.48	.77	75	.67	65.1	.50
Union	**88.58**	**.95**	85.34	.83	65.1	.50

Given that our Random forest classifier performed better in general for entity mentions, we based our following analysis on its results (Table 11). For the *correct* label, the use of TAGME_Mention or Union provided the highest f1-score (.83), but the use of TAGME_Mention alone provided slightly better precision (.87). On the *not-correct* label, once again, TAGME_Mention and the Union achieved the highest f1-score (.91), but this time the Union alone gave slightly better precision (.90).

Table 11. Precision, recall & f1-score using Entity mentions

Label	Classifier	Tool	Precision	Recall	F1
Correct	Random Forest	Spotlight Mention	.85	.47	.61
		TAGME Mention	**.87**	.79	**.83**
		Intersection	.84	.48	.61
		Union	.86	**.80**	**.83**
	SVM	Spotlight Mention	.77	.40	.53
		TAGME Mention	**.81**	.75	**.78**
		Intersection	.77	.40	.53
		Union	**.81**	**.76**	**.78**
	ZeroR	Spotlight Mention	0	0	0
		TAGME Mention	0	0	0
		Intersection	0	0	0
		Union	0	0	0
Not-correct	Random Forest	Spotlight Mention	.77	**.96**	.85
		TAGME Mention	.89	.94	**.91**
		Intersection	.77	**.95**	.85
		Union	**.90**	.93	**.91**
	SVM	Spotlight Mention	.75	**.94**	.83
		TAGME Mention	**.87**	.91	**.89**
		Intersection	.74	**.93**	.83
		Union	**.87**	.91	**.89**
	ZeroR	Spotlight Mention	.65	1	.79
		TAGME Mention	.65	1	.79
		Intersection	.65	1	.79
		Union	.65	1	.79

An explanation for the difference in performance between Spotlight_Mention and TAGME_Mention is the amount of mentions retrieved by each of the semantic annotators. Spotlight provided fewer annotations for the same answers than TAGME. In addition, our manual inspection of annotations revealed that TAGME tended to produce more accurate annotations than Spotlight. This suggests that higher quantity and quality of semantic annotations leads to a feature set that successfully differentiates between *correct* and *not-correct* answers.

5.3 Entity URI Results

The results presented in Table 12 show that Random forest provided highest accuracy and AUC on each configuration. The best accuracy and AUC were achieved by Random forest with TAGME_URI (86.60% and .94, respectively).

Table 12. Accuracy & AUC using Entity URIs

Tool	Random Forest		SVM		ZeroR	
	ACC %	AUC	ACC %	AUC	ACC %	AUC
Spotlight URI	80.55	.84	77.60	.75	60.8	.45
TAGME URI	**86.60**	**.94**	84.74	.82	65.1	.45
Intersection	77.03	.82	76.44	.74	59	.45
Union	86.50	.94	82.80	.80	63.6	.45

We notice that in terms of accuracy and AUC, TAGME_URI and Union on Random forest are slightly lower than TAGME_Mention and Union for Entity mention features.

Focusing on Random forest as the best performing classifier, we observe that for the *correct* label, the use of TAGME_URI and union of entity URIs provided the best f1-score of .80 (Table 13). In terms of precision, the union of entity URIs had a better performance (.86). For the *not-correct* label, again on Random forest, TAGME_URI and the Union configurations get better f1-score (.90). This time TAGME_URI alone provided the best precision (.88) for this label.

We observed that in some cases, the same mention was associated to different entity URIs in two different answers and that only one of the URIs was correct. When this happens, it affects the quality of the vector representation of student answers by increasing the number of URIs in the VSM vocabulary, thus making the representation even sparser.

Table 13. Precision, recall & f1-score using Entity URIs

Label	Classifier	Tool	Precision	Recall	F1
Correct	Random Forest	Spotlight URI	.82	.64	.72
		TAGME URI	.84	**.76**	**.80**
		Intersection	.77	.63	.69
		Union	**.86**	**.76**	**.80**
	SVM	Spotlight URI	.77	.62	.68
		TAGME URI	**.81**	**.73**	**.77**
		Intersection	.77	.61	.68
		Union	.80	.71	.75
	ZeroR	Spotlight URI	0	0	0
		TAGME URI	0	0	0
		Intersection	0	0	0
		Union	0	0	0
Not-correct	Random Forest	Spotlight URI	.80	.91	.85
		TAGME URI	**.88**	.92	**.90**
		Intersection	.77	.87	.82
		Union	.87	**.93**	**.90**
	SVM	Spotlight URI	.78	.88	.83
		TAGME URI	**.86**	**.91**	**.89**
		Intersection	.76	.87	.81
		Union	.84	.90	.87
	ZeroR	Spotlight URI	.61	1	.76
		TAGME URI	**.65**	1	**.79**
		Intersection	.59	1	.74
		Union	.64	1	.78

5.4 Entity Embedding Results

Among models trained using entity embeddings, the highest accuracy and AUC were achieved by Random forest with the TAGME_URI configuration, as shown in Table 14. For this feature set, we observe that Random forest has higher accuracy with TAGME_URI and Union than SVM on the same configurations; but SVM gets higher accuracy than Random forest using Spotlight_URI and Intersection. However, the AUC for Random forest is still higher than for SVM in all the configurations. We can also observe an increase in accuracy and in AUC (although modest) for the baseline.

Table 14. Accuracy & AUC using Entity embeddings

Tool	Random Forest		SVM		ZeroR	
	ACC %	AUC	ACC %	AUC	ACC %	AUC
Spotlight URI	80.13	.86	81.13	.71	73.5	.50
TAGME URI	**82.67**	**.90**	75.46	.70	63.7	.50
Intersection	76.43	.81	79.64	.66	74.6	.49
Union	82.45	.89	80.79	.69	73.5	.50

Further inspection of the results obtained on cross-validated models (Table 15) reveals that this time, the highest results differ between classification algorithms. For the *correct* label, we obtained better f1-score with Random forest using the union of entity embeddings (.89). However, SVM provided better precision using Spotlight (.88). The *not-correct* label had both the best precision (.85 using the union of entity embeddings) and f1-score (.87 using TAGME_URI) results using Random forest.

Table 15. Precision, recall & f1-score using Entity embeddings

Label	Classifier	Tool	Precision	Recall	F1
Correct	Random Forest	Spotlight URI	.83	.92	.87
		TAGME URI	**.85**	.64	.73
		Intersection	.81	.90	.85
		Union	.82	**.97**	**.89**
	SVM	Spotlight URI	**.88**	.92	**.88**
		TAGME URI	.74	.50	.60
		Intersection	.82	.93	**.88**
		Union	.82	**.94**	**.88**
	ZeroR	Spotlight URI	.73	1	.85
		TAGME URI	0	0	0
		Intersection	**.75**	1	**.86**
		Union	.73	1	**.86**
Not-correct	Random Forest	Spotlight URI	.68	.47	.56
		TAGME URI	.82	**.93**	**.87**
		Intersection	.56	.37	.44
		Union	**.85**	.41	.55
	SVM	Spotlight URI	.70	.50	.58
		TAGME URI	**.76**	**.90**	**.82**
		Intersection	.67	.39	.50
		Union	.72	.45	.55
	ZeroR	Spotlight URI	0	0	0
		TAGME URI	**.64**	**1**	**.78**
		Intersection	0	0	0
		Union	0	0	0

Even though TAGME_URI provided better precision for the correct label, the union of entity embeddings got better f1-score and recall. The increase in f1-score can be related to the amount of entity URIs provided by the Union (set union of entity URIs from DBpedia Spotlight and TAGME). This suggests that more entities have a positive effect on performance. Similarly, as in accuracy, there was an increase in precision and f1-score on both labels for the baseline classifier.

5.5 Mention Embedding Results

Table 16 shows that when mention embeddings were used as features, SVM achieved the highest accuracy of 81.79% with the Union configuration. This is the first time that Random forest is surpassed by SVM in terms of accuracy, However, Random forest is still outperforming SVM in terms of AUC.

Table 16. Accuracy & AUC using mention embeddings

Tool	Random Forest		SVM		ZeroR	
	ACC %	AUC	ACC %	AUC	ACC %	AUC
Spotlight Mention	74.83	.74	75.83	.63	73.50	.50
TAGME Mention	80.85	.85	79.66	.76	63.69	.50
Intersection	78.50	.73	79.52	.68	74.40	.48
Union	79.80	**.86**	**81.79**	.71	73.50	.49

As in entity embeddings, the highest results differ between classification algorithms. Table 17 presents detailed results for the performance of Random forest and SVM using mention embeddings. For the *correct* label, the Random forest classifier with the union of mention embeddings had f1-score of .88 (the highest F1 value). For precision, SVM did better with either the intersection or union of mention embeddings (.83). The *not-correct* label had both best precision (.86 using TAGME_Mention) and f1-score (.83 using the union of mention embeddings) with the SVM classifier.

Table 17. Precision, recall & f1-score using mention embeddings

Label	Classifier	Tool	Precision	Recall	F1
Correct	Random Forest	Spotlight Mention	.78	.91	.84
		TAGME Mention	**.82**	.61	.70
		Intersection	.81	.93	.87
		Union	.80	**.98**	**.88**
	SVM	Spotlight Mention	.80	.90	.85
		TAGME Mention	.78	.61	.69
		Intersection	**.83**	.92	.87
		Union	**.83**	**.94**	**.88**
	ZeroR	Spotlight Mention	.73	1	.85
		TAGME Mention	0	0	0
		Intersection	**.74**	1	.85
		Union	.73	1	.85
Not-correct	Random Forest	Spotlight Mention	.55	.30	.39
		TAGME Mention	.81	**.92**	**.86**
		Intersection	.64	.36	.46
		Union	**.83**	.30	.44
	SVM	Spotlight Mention	.57	.36	.44
		TAGME Mention	**.80**	**.90**	**.85**
		Intersection	.65	.44	.52
		Union	.75	.48	.58
	ZeroR	Spotlight Mention	0	0	0
		TAGME Mention	**.64**	**1**	**.78**
		Intersection	0	0	0
		Union	0	0	0

6. FEATURE SELECTION

In this section, we describe the results obtained when applying two feature selection methods to our dataset: mean decrease impurity (MDI) and mean decrease accuracy (MDA). Both methods employ random trees to measure the importance of a feature [7]. We trained different classifiers with the selected features and compared their results to the same classifiers without feature selection.

First, we calculated the MDA and MDI scores for each feature in our data set and kept only features with scores strictly higher than 0. Negative or zero MDA/MDI values were either detrimental or unhelpful to the performance of the classifiers. Table 18 shows the number of features before and after feature selection.

Table 18. Number of remaining features with and without (WFS) feature selection

	Technique		
Features	**WFS**	**MDA**	**MDI**
N-gram	700	90	205
Entity Mention	665	99	179
Entity URI	875	109	236
Entity Embedding	100	83	84
Mention Embedding	300	161	117

Then, we trained and evaluated Random Forest, SVM, and ZeroR classifiers using each of the top performing configurations per feature set in terms of f1-score to compare the results obtained with and without feature selection. The obtained results (Figure 1) show that in most cases, feature selection led to a slight increase in the accuracy of our classifiers. Specifically, MDA improved the accuracy of the classifiers in every case by as much as 4.9 for SVM using mention embeddings as features. However, overall Random forest generally remained the best, in terms of accuracy, with and without feature selection.

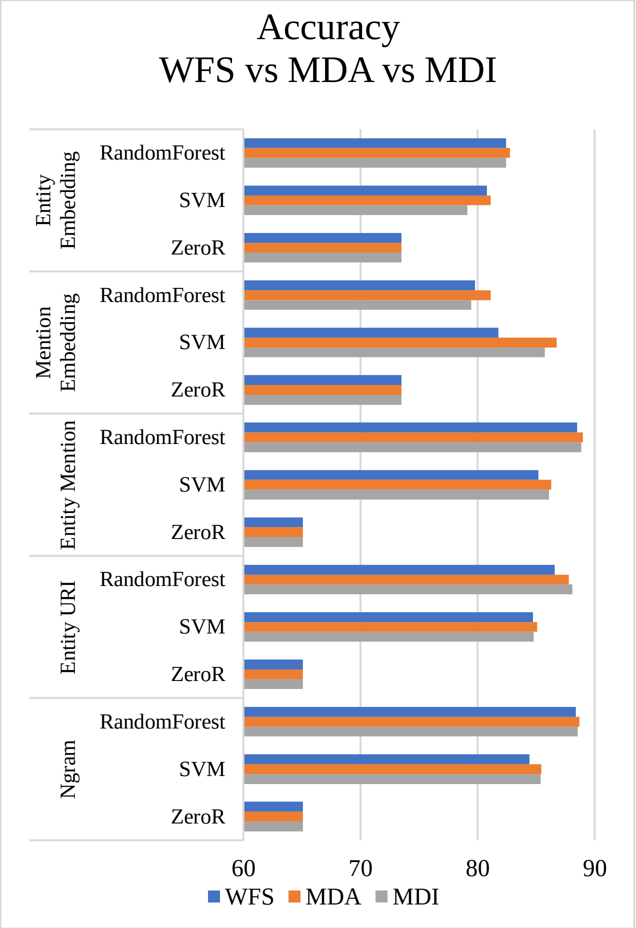


Figure 1. Accuracy without feature selection (WFS) versus MDA & MDI

7. DISCUSSION

Overall, our Random forest classifiers proved the best in terms of accuracy and AUC. The only exception is with mention embeddings in which SVM did better in terms of accuracy by at most 1 percentage point. Therefore, we base our conclusions only on Random forest.

In terms of accuracy, there was not much difference between several feature sets as shown on Figure 2. The two best feature sets for accuracy were entity mentions with the union (88.58%) or TAGME configurations (88.52%) and n-gram features with the unigrams configuration (88.40%); these feature sets achieved the highest AUC (.95), as well.

In terms of precision (Figure 3), n-grams outperformed other feature sets for the *correct* label (.93) and entity mentions obtained the best results for the *not-correct* label using the union and TAGME configurations (.90, .89).

For F1-score (Figure 4), entity embeddings achieved the highest score for the *correct* label (.89 using the union configuration) closely followed by mention embeddings (union). Entity mentions (using the union or TAGME configurations) and unigrams did better for *not-correct* (.91) followed by entity URIs (.90 with TAGME and union) and n-grams.

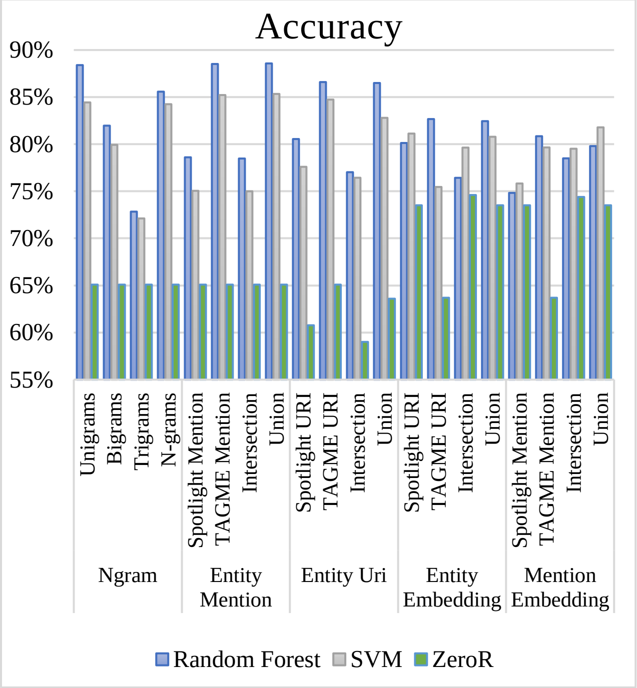


Figure 2. Accuracy results

When considering which class (*correct*, *not-correct*) we were best able to predict in terms of precision (Figure 3), we found that the detection of *correct* answers was better than *not-correct answers*, with differences ranging from .01 to .25 with Random forest. N-gram features were better at detecting *correct* answers than *not-correct* ones; while entity mentions did better for the *not-correct* (using union or TAGME) label. In 14 out of our 20 possible configurations, the classifiers were more precise in detecting *correct* answers. This is the case despite the unbalanced ratio of 35% *correct* answers and 65% *not-correct* answers used for training. When we focus on the f1-score (Figure 4) we obtain better results for the *not-correct* label. We observe that the union configuration for entity embeddings and mention embeddings is the best for *correct* answers while entity mentions (TAGME or union) followed by unigrams outperform the other features for the *not-correct* answers.

On average, unigrams are the best at differentiating between correct and not-correct labels in terms of precision while entity mentions (either with TAGME or Union) is preferred in terms of f1-score.

The best configuration based on semantic annotations depends on the considered evaluation metric. Based on accuracy, features that use mentions (entity mentions and mention embeddings) performed better with either union or TAGME. The feature sets that use URIs (entity URIs and entity embeddings) performed better with URIs obtained using TAGME. In both cases, the use of TAGME alone obtains either the best result or is very close to the highest value. For f1-score, the use of TAGME for entity mentions and entity URIs provided the same results as the union for both labels; additionally, TAGME and union are also the best configurations for both entity mentions and entity URIs. Entity embeddings and mention embeddings had their best f1-score on the *correct* label using the union, but better f1-score for *not-correct* using TAGME alone. When we average the f1-score for both labels, we obtain higher results with TAGME. The reason for very similar results with TAGME and the union is that the annotations provided by Spotlight were often a subset of those provided by TAGME.

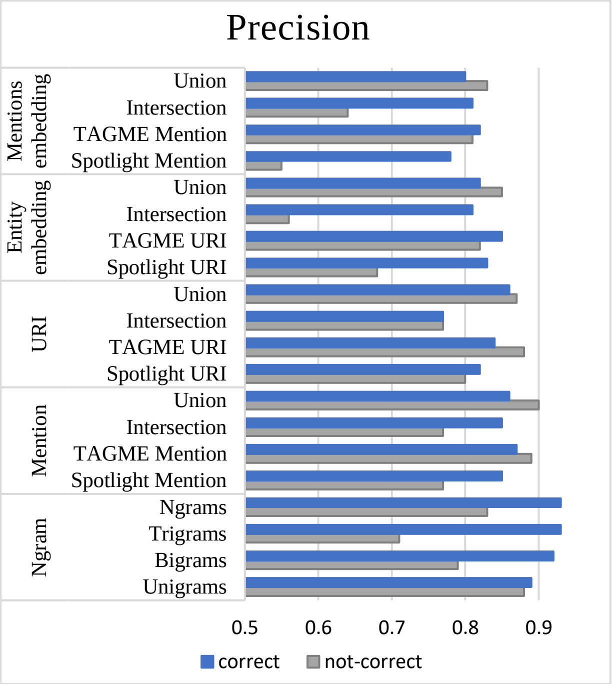


Figure 3. Precision results for Random forest

Both entity and mention embeddings performed worse than n-gram features and semantic annotations models based on accuracy. However, one interesting observation is that, for the *correct* label, entity and mention embeddings outperformed all features on f1-score (Figure 4). Entity embeddings obtained slightly better results (precision, f1-score and accuracy) compared to mention embeddings.

Our feature selection efforts show that MDI did not consistently improve the overall accuracy of our classifiers. It was the MDA feature selection technique which provided improvement in all the cases. The increase in accuracy ranged from .3% to 4.9%.

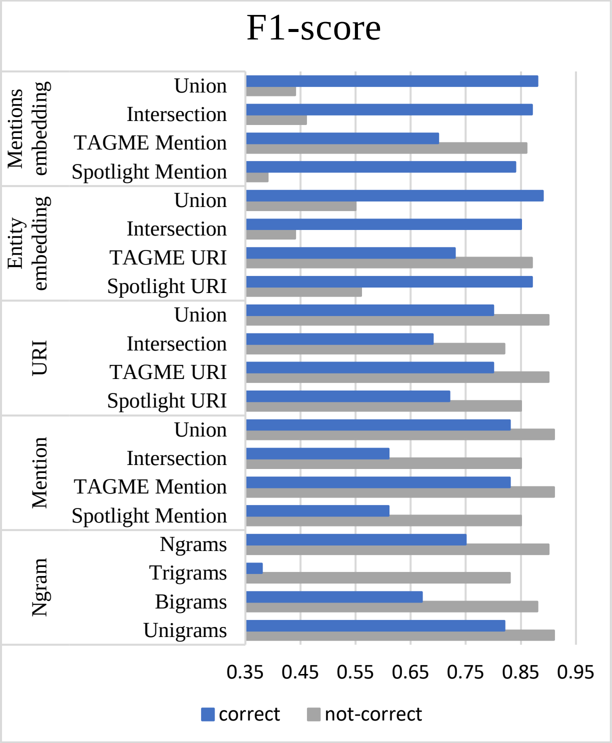


Figure 4. F1-score results for Random Forest

8. CONCLUSION

In this paper, we compared several vector-based feature sets coupled with classifiers for the ASAG task.

In general, we showed that on average, entity mention features (TAGME or union) are the top features in terms of f1-score while n-gram features (unigrams) are the best in terms of precision. For the detection of *correct* answers, we showed that n-gram features (trigrams and n-grams) and features based on embeddings (entity and mention embeddings with the union configuration) are the most effective in terms of precision and f1-score respectively. In terms of semantic annotations, TAGME provided the best accuracy for each feature with the exception of entity mentions, where the union configuration slightly outperformed TAGME alone. Finally, the MDA feature selection technique slightly improved the accuracy of all the classifiers.

One of the main limitations of this study is the unbalanced set of labeled answers available in the corpus. Another limitation is associated with the configuration of semantic annotators as we only tested the default level of confidence for each annotator. One additional limitation, for mention embeddings specifically, is the relatively low coverage obtained using GloVe. We plan to address these limitations in future work by testing the proposed features against other available ASAG datasets. We also intend to experiment with varying the level of confidence and similar parameters of the semantic annotators. Another important step will be to exploit a combination of the current features to benefit from their respective strengths for the correct and not correct labels. Finally, we will explore other methods for response classification using additional features that exploit model answers and deep learning architectures.

9. ACKNOWLEDGMENTS

This research is supported by an INSIGHT grant from the Social Sciences and Humanities Research Council of Canada (SSHRC).

10. REFERENCES

[1] Basu, S., Jacobs, C., and Vanderwende, L. 2013. Powergrading: a clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics*, 1, 391-402.

[2] Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. 2007. Greedy layer-wise training of deep networks. In *Proceedings of Advances in Neural Information Processing Systems Conference*, 153-160.

[3] Biggs, J., and Tang, C. 2011. *Teaching for quality learning at university (4th ed.).* London, UK: McGraw-Hill International.

[4] Burrows, S., Gurevych, I., and Stein, B. 2015. The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, 25, 1, 60–117.

[5] Dessus, P., Lemaire, B., and Vernier, A. 2000. Free-text assessment in a virtual campus. In *Proceedings of the 3rd International Conference on Human System Learning*, 61-75.

[6] Ferragina, P., and U. Scaiella. 2010. TAGME: On-the-fly Annotation of Short Text Fragments (by Wikipedia Entities). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, 1625–1628.

[7] Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. 2013. Understanding variable importances in forests of randomized trees. In *Proceedings of Advances in Neural Information Processing Systems Conference*. 431–439.

[8] Heilman, M., and Madnani, N. 2013. ETS: Domain adaptation and stacking for short answer scoring. In Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval)*, 2, 275-279.

[9] Jayashankar, S., and Sridaran, R. 2017. Superlative model using word cloud for short answers evaluation in eLearning. *Education and Information Technologies*, 22, 5, 2383-2402.

[10] Jordan, S., and Mitchell, T. 2009. e-Assessment for learning? The potential of short-answer free-text questions with tailored feedback. *British Journal of Educational Technology*, 40, 2, 371–385.

[11] Klein, R., Kyrilov, A., and Tokman, M. 2011. Automated assessment of short free-text responses in computer science using latent semantic analysis. In G. Roßling, T. Naps, C. Spannagel (Eds.), In *Proceedings of the 16th annual joint conference on innovation and technology in computer science education*, 158–162. Darmstadt: ACM. (CAPS'3), 61-76.

[12] Leacock, C., and Chodorow, M. 2003. C-rater: automated scoring of short-answer questions, *Computers and the Humanities*, 37, 4, 389–405.

[13] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, Sö. and Bizer, C. 2015. DBpedia - A Large-scale,

Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*, 6, 167-195.

[14] Madnani, N., Loukina, A., and Cahill, A. (2017). A Large Scale Quantitative Exploration of Modeling Strategies for Content Scoring. *In Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, 457-467.

[15] Magooda, A. E., Zahran, M. A., Rashwan, M., Raafat, H. M., and Fayek, M. B. 2016. Vector Based Techniques for Short Answer Grading. In *Proceedings of Florida Artificial Intelligence Research Society Conference* (*FLAIRS)*, 238-243.

[16] McDonald, J., Bird, R. J., Zouaq, A., and Moskal, A. C. M. 2017. Short answers to deep questions: supporting teachers in large-class settings, *Journal of Computer Assisted Learning*. 33, 4 306–319.

[17] McDonald, J., Knott, A., and Zeng, R. 2012. Free-text input vs menu selection: exploring the difference with a tutorial dialogue system. In *Proceedings of the Australasian Language Technology Association Workshop 2012*, 97-105.

[18] McDonald, J., Knott, A., Zeng, R., and Cohen, A. 2011. Learning from student responses: A domain-independent natural language tutor. In *Proceedings of the Australasian Language Technology Association Workshop*, 148-156.

[19] Mendes, P. N., Jakob, M., García-Silva, A., and Bizer, C. (2011, September). DBpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, ACM, 1-8.

[20] Mikolov, T., Chen, K., Corrado, G. and Dean, J. 2013. Efficient Estimation of Word Representations in Vector Space, In *Proceedings of Workshop at International Conference on Learning Representations ICLR.*

[21] Oren, E, Moller K, Scerri, S, Handschuh, S and Sintek, M 2006, What are Semantic Annotations? *Relatório técnico. DERI Galway*, 9, 62-75.

[22] Papineni, K., Roukos, S., Ward, T., and Zhu, W. J. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 311-318.

[23] Pennington, J., Socher, R., and Manning, C. 2014. GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532-1543.

[24] Reilly, E. D., Stafford, R. E., Williams, K. M., and Corliss, S. B. 2014. Evaluating the validity and applicability of automated essay scoring in two massive open online courses, *The International Review of Research in Open and Distributed Learning*, 15, 5, 83-98.

[25] Roy, S., Bhatt, H. S., and Narahari, Y. 2016. Transfer Learning for Automatic Short Answer Grading. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI)*, Hague, Netherlands, 1622-1623.

[26] Riordan, B., Horbach, A., Cahill, A., Zesch, T., and Lee, C. M. 2017. Investigating neural architectures for short answer scoring. *In Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, 159-168.

[27] Sakaguchi, K., Heilman, M., and Madnani, N. 2015. Effective feature integration for automated short answer scoring. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1049-1054.

[28] Shermis, M. D. 2015. Contrasting state-of-the-art in the machine scoring of short-form constructed responses. *Educational Assessment*, 20, 1, 46-65.

[29] Tack, A., François, T., Roekhaut, S., and Fairon, C. 2017. Human and Automated CEFR-based Grading of Short Answers. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, 169-179.

[30] The Hewlett Foundation. 2012. The Automated Student Assessment Prize: Short Answer Scoring. http://www.kaggle.com/c/asap-sas

[31] Wolpert, D. H. 1992. Stacked generalization. Neural Networks, 5, 2, 241–259.

[32] Yuan, L., and Powell, S. 2013. MOOCs and open education: Implications for higher education. Centre for Educational Technology & Inoperability Standards. Retrieved from http://publications.cetis.ac.uk/wp-content/uploads/2013/03/MOOCs-and-Open-Education.pdf

[33] Zhang, Y., Shah, R., and Chi, M. 2016. Deep Learning + Student Modeling + Clustering: a Recipe for Effective Automatic Short Answer Grading. In *Proceedings of the 9th International Conference on Educational Data Mining* (*EDM)*, 562-567.

[34] Zhou, H., Zouaq, A., and Inkpen, D. 2017. DBpedia Entity Type Detection Using Entity Embeddings and N-Gram Models. In *Proceedings of International Conference on Knowledge Engineering and the Semantic Web*, 309-322.

Behavioral Analysis at Scale: Learning Course Prerequisite Structures from Learner Clickstreams

Weiyu Chen
Advanced Research
Zoomi Inc.
weiyu.chen@zoomiinc.com

Andrew S. Lan
Department of Electrical Engineering
Princeton University
andrew.lan@princeton.edu

Da Cao
Advanced Research
Zoomi Inc.
da.cao@zoomiinc.com

Christopher Brinton
Advanced Research
Zoomi Inc.
chris.brinton@zoomiinc.com

Mung Chiang
College of Engineering
Purdue University
chiang@purdue.edu

ABSTRACT

Knowledge of prerequisite dependencies is crucial to several aspects of learning, from the organization of learning content to the selection of personalized remediation or enrichment for each learner. As the amount of content is scaled up, however, it becomes increasingly difficult to manually specify all of the prerequisites among the different content parts, necessitating automation. Since existing approaches to automatically inferring prerequisite dependencies rely on analysis of content (*e.g.,* topic modeling of text) or performance (*e.g.,* quiz results tied to content) data, they are not feasible in cases where courses have no assessments or only short content pieces (*e.g.,* short video segments). In this paper, we propose an algorithm that extracts prerequisite information using learner *behavioral data* instead of content and performance data, and apply it to an online short course. By modeling learner interaction with course content through a recurrent neural network-based architecture, our algorithm characterizes the prerequisite structure as latent variables, and estimates them from learner behavior. Through evaluation on a dataset of roughly 12,000 learners in a course we hosted on our platform, we show that our algorithm excels at both predicting behavior and revealing fine-granular insights into prerequisite dependencies between content segments, with validation provided by a course administrator. Our approach of content analytics using large-scale behavioral data complements existing approaches that focus on course content and/or performance data.

1. INTRODUCTION

Recent advances in machine learning and big data have provided opportunities to revamp the traditional "one-size-fits-all" approach to education. Researchers have developed methods that analyze massive learner and content data to provide personalized recommendations on what actions learners should take, e.g., to read a section of a textbook, watch a lecture video, or work on a practice question [19, 24]. By catering to the needs of each individual learner, such personalization methods can enhance learning efficacy; see [1] for an overview.

By specifying an ordering of which learning content should be used before others, content prerequisite structures provide important guidance for the design of personalization algorithms. These structures may be defined at multiple levels of granularity, from across courses to within single pieces of learning content (e.g., between chunks of a video), or for specific units of knowledge (often termed "knowledge components", "skills", or "concepts"). Roughly speaking, learning content is deemed the prerequisite of another if it contains knowledge that learners have to master before studying the other. For example, Calculus is a prerequisite of Differential Equations at the granularity of different courses; learners should master the former before they learn the latter.

Several works have demonstrated the utility of prerequisite structures to learning and personalization. For one, [32] showed that when instructors do not take these prerequisite structures into account when designing their course curriculums, learners do not perform as well. Also, [33] showed that learners with high mastery of prerequisite knowledge are much less likely to become confused in learning tasks, compared to those with low mastery. Moreover, the works in [4, 37] showed that an important feature in the prediction of a learner's first responses on a particular skill is the learner's demonstrated mastery level on prerequisite skills. But existing methods for extracting prerequisites suffer from important drawbacks that we will describe next.

1.1 Existing Methods for Prerequisite Structure Extraction

Explicit prerequisite structures, like those in [32], are labor-intensive to construct manually and rarely available in practice, especially when considering fine-granular prerequisites (e.g., between file segments). Inexplicit structures on the other hand, such as tables of contents in textbooks [18] and knowledge graphs constructed from large databases [3], typically only contain weak information about prerequisites: they offer some information on how learning content should be ordered, but do not necessarily impact learner performance or behavior. This observation has motivated the development of automated methods for extracting explicit prerequisite structures from data. Existing methods of automation can be divided into two main categories based on the type of data they use: (i) learner data and (ii) content data.

Methods in the first category use one form of learner data almost exclusively: learner performance, which usually consists of learners' responses to assessment/quiz questions. These methods have used several different models/algorithms to make inferences from performance data, including causal graphs [28], structural expectation-maximization [9], Bayesian estimation [14], hypothesis testing [6], probabilistic association rules [10], convex optimization [27], correlation/regression analysis [7], and approximate Kalman filtering [21].

As for the second category, methods have leveraged several forms of content data and metadata. [18], for instance, proposed using the organization and unit titles in online textbooks to classify between prerequisite and outcome concepts. Others have involved Wikipedia, either using the content on wiki pages to aid the extraction of concept maps in textbooks [34,35] or extracting prerequisite structures among the pages themselves [22, 31]. While [22] analyzed the links between pages, [31] uses both textual content and the page creation and modification logs to extract prerequisites.

The major downside of these existing automation methods is that they require substantial learner performance or content data, which is not always available or accessible. Corporate training, for example, is a learning scenario in which many courses have few if any assessments; performance is in many cases assigned as a single satisfactory/unsatisfactory outcome at the end of the course [8]. Methods that extract prerequisite structures based on learner performance data, then, are not applicable in these settings. On the other hand, in many interactive learning environments like educational games [23], content data is limited and not easily parsable; in these settings, methods to infer prerequisites based on content data (especially text) are not applicable. Moreover, in any learning scenario, as the level at which prerequisites are desired becomes more fine-grained, the amount of content data available in each content piece becomes smaller.

As a result, there is a need to develop methods that can extract prerequisite structures from sources of data that (i) are abundant in different learning scenarios and (ii) can be captured within fine-granular pieces of content, especially in settings where content and performance data are limited.

1.2 Our Method and Contributions

In this paper, we develop the first methodology to extract prerequisite structures from large-scale learner *behavioral* data, using a novel recurrent neural network (RNN)-based probabilistic model. Behavioral data measures learner interaction with course material, typically in the form of clickstream logs that are generated based on each mouse click; in this way, it can be captured on small pieces of content in any online learning scenario. We demonstrate the ability of our model to identify prerequisites between fine-granular content segments in the setting of online short-courses, where performance and content data are limited; for our particular dataset, the entire course is less than 15 minutes in duration, and while the 12,000 learners do not respond to any assessment questions, they generate almost 900,000 clickstreams.

Specifically, our methodology consists of three main steps:

Feature engineering. First, we analyze the behavioral data captured by our online learning platform in terms of a set of learning features (Section 2). These features summarize a learner's behavior on each segment of content that they visit as one of four states: low or high engagement if they studied the segment, and skipping back or forward otherwise. In deriving the formulas to convert from data to features, we consider cases of off-task behavior (e.g., idle time) that should be filtered out. We also consider content features in our model; since the content data is sparse, we embed each segment according to pre-trained statistical language models.

Modeling and inference. Second, we infer the parameters of our probabilistic model through training and validation on the dataset. The RNN-based learner model we propose (Section 3) consists of two main parts: (i) a latent knowledge state transition model, which considers how a learner's knowledge state changes based on the segment visited and behavior exhibited, and (ii) a learner behavior model, which characterizes the probability that the learner exhibits a particular behavior based on their current knowledge gaps. Our model parameters are trained by minimizing cross-entropy loss in the prediction of learner behavior on segments they visit.

Prerequisite analysis. Third, we analyze prerequisite information for our dataset by examining a model parameter matrix that specifies dependencies between segments (20 second chunks of video in this course). To establish reliability, we start by evaluating the performance of our model in predicting behavior on our dataset (Section 4.2); in doing so, we find that it can obtain over 85% accuracy and significant improvements over baselines. Then, we visualize the prerequisite matrix, discuss its insights it provides, and verify them through a questionnaire provided to a course administrator (Section 4.3).

At the end, we also describe how our model parameters can drive content personalization. More generally, we believe that this work will motivate a new research thrust in using human behavior to aid content analytics: such approaches have the potential to benefit applications that involve large-scale human-content interaction but have only limited content data.

2. BEHAVIORS AND CONTENT: DATA AND FEATURES

In this section, we detail our methods for processing learner behavioral data. We first discuss the specific course dataset we consider, then the data capture, and finally the computation of features from this data that are used in our prerequisite identification algorithm.

2.1 Course and Enrollment

The dataset we use comes from an online course on the topic of product development that we hosted on our course delivery platform. This course consists of 4 sequential videos that we divide into a total of 36 segments, with each segment spanning 20 seconds; totaling less than 15 minutes, this qualifies as a short-course [8].

We let $s = 1, 2, ..., S$ denote the index of the segments in the course sequence. Our evaluation will focus on the roughly 12,000 learners who enrolled in this course over a six-month period in 2017.

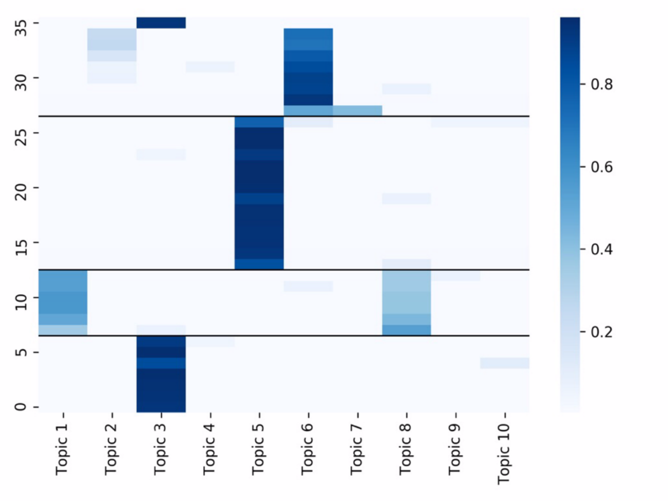


Figure 1: Visualization of the topic distributions across video segments in the course, as inferred by LDA. We see that videos tend to cover disparate sets of topics; therefore, this analysis does not help us to extract prerequisite structures.

2.2 Data Capture

We focus on two types of data captured by the platform: (i) video-watching clickstreams, which log each learner's interactions with the video player, and (ii) transcripts of the course content, measured in words. In total, this data consists of roughly 900,000 clickstreams and 1,700 words across the video segments.

Given such a limited text repository, relying on topic models alone to extract prerequisite structures is infeasible. Nonetheless, we incorporate content data as one component of our methodology, since we seek to use any data sources available to aid the performance of our model. In later sections, we will experimentally validate the impact of this input on model performance, and the possibility of replacing it with other data.

Video-watching clickstreams. The data capture architecture for our platform is event-driven, i.e., each event that a learner makes is recorded. The following is the space of actions available to a learner on the video scrub bar: Play (`Pl`), Pause (`Pa`), Skip forward (`Sf`), and Skip backward (`Sb`). There are also actions available outside of the scrubber: Enter video (`En`), Exit video (`Ex`), Window foreground (`Wf`), and Window background (`Wx`), where `Wf` and `Wx` dictate whether the course application is the current selection on the device. Formally, the ith event created by learner u in the course will be in the format

$$E_u(i) = < v(i), a(i), s'(i), s(i), p(i), b(i) >,$$

where $v(i)$ is the video ID and $a(i)$ is the type of action. $s(i)$ is the segment of the video player immediately after $e(i)$ was fired, while $s'(i)$ is the one immediately before. $p(i)$ is the UNIX timestamp (in seconds) of this event, and $b(i) \in \{\texttt{playing}, \texttt{paused}\}$ is the binary state of the video player immediately after i happens.

For a video with multiple segments, when the learner plays through the end of s, an event with $a(i) = \texttt{play}$, $s'(i) = s$, and $s(i) = s+1$ will be generated.

Course content. The videos originate in .mp4 format for delivery to learners. To obtain the text transcripts, we divide videos to length of 20-second long segments and employ open source speech-to-text conversion software, creating one output for each segment and further correcting any translation mistakes manually. Concretely, the output for segment s in the bag-of-words representation $\mathbf{x}_s$ over a dictionary $\mathscr{X} = \{w_1, w_2, ...\}$, where $\mathbf{x}_s(k)$ is the number of times word $w_k \in \mathscr{X}$ appears in s.

To further motivate our behavior-based approach to inferring prerequisites, in Figure 1 we show the progression of topics through the segments in the course as inferred by the latent Dirichlet allocation (LDA) topic analysis algorithm [2]. LDA extracts document-topic and topic-word distributions from a corpus of text separated into documents; here, segments are treated as separate documents, and the segment-topic distributions are plotted. According to this model, each video focuses on fairly independent topics, with minimal overlap (e.g., the segments in the first video focus heavily on topic 3, while those in the third focus almost entirely on topic 5). This analysis shows how topic analysis alone provides limited insights into prerequisite structures which likely extend across videos, a point we will verify later in our model evaluation.

2.3 Feature Construction

We construct two types of features from our data: (i) video-watching behaviors and (ii) text embedding vectors. The behaviors are learner-specific, while the text vectors are not.

Video-watching behaviors. Let $s(u,t)$ denote the segment learner u visited at time index $t \in \{1, ..., T_u\}$, with T_u being the total number of (not necessarily unique) segments u visited. The time instance here increments whenever the learner transitions to a different segment, i.e., $s(u,t) \neq s(u,t+1)$. In our model, we consider the behavior of learner u at time t as a feature $f_{u,t} \in \mathscr{F}$, where $\mathscr{F} = \{\texttt{LE}, \texttt{HE}, \texttt{SB}, \texttt{SF}\}$ is a set of four states summarizing behavior on a segment: Low Engagement (`LE`), High Engagement (`HE`), Skip Back (`SB`), and Skip Forward (`SF`).

$f_{u,t}$ is determined by analyzing the set of measurements $E_{u,t}$ that occur for learner u during time t. Letting $i(t)$ and $i(t+1)$ be the indices of the events where u transitions to $s(u,t)$[1] and $s(u,t+1)$, respectively, then $E_{u,t} = \{E_u(i) : i(t) \leq i < i(t+1)\}$. From this, we first calculate the time spent on $s(u,t)$ by aggregating the changes in timestamps between sequential events in $E_{u,t}$, excluding any points of the app in the background that indicate learner off-task behavior:

$$m_{s(u,t)} = \sum_{\substack{i,i+1 \in E_{u,t} \\ a(i) \neq \texttt{Wx}}} \min\left(p(i+1) - p(i), T_u\right),$$

where $T_u = 300$ sec is an upper bound for idle time on each 20 second segment.

If $m_{s(u,t)} < 3$, then we infer that the learner has skipped over $s(u,t)$; in this case, if $s(u,t+1) > s(u,t)$, then it is a forward skip and $f_{u,t} = \texttt{SF}$, whereas if $s(u,t+1) < s(u,t)$ then it is backwards and $f_{u,t} = \texttt{SB}$. On the other hand, if $m_{s(u,t)} \geq 3$, then the learner has engaged with the segment; similar to [8], we quantify engagement on $s(u,t)$ as

$$e_{s(u,t)}(m) = \left(\frac{1 + m_{s(u,t)}/\bar{m}_s}{2}\right)^{\alpha},$$

[1] in other words, $i(t) = i : s'(i) \neq s(u,t), s(i) = s(u,t)$.

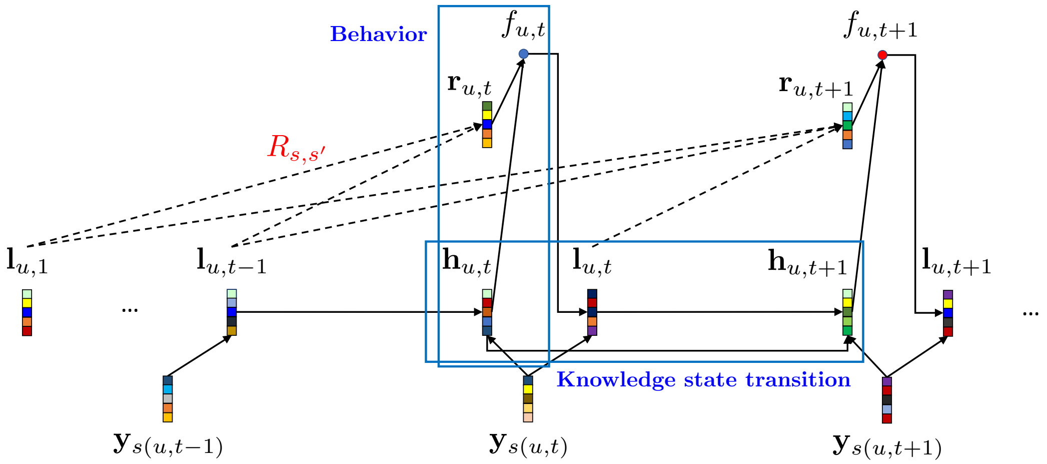


Figure 2: Roll-out visualization of the architecture of our RNN-based learner behavior model. At time $t+1$, we use the observed learner behavior $f_{u,t}$ at time t, the learner's prior knowledge state $h_{u,t}$, and the knowledge contained in the previous segment $s(u,t)$ to update their current knowledge state. Then, we calculate the prerequisite knowledge gap and the learning goal knowledge gap using the prerequisite structure $\mathbf{R}$ among segments, which decide the learner's behavior at time $t+1$. Latent variable dependencies are denoted by solid arrows, while the prerequisite dependencies are denoted by dashed arrows.

where $\bar{m}_s$ is the expected time spent on s, and $\alpha \in (0,1]$ is a parameter for the diminishing marginal returns of time spent on engagement.[2] Intuitively, $m_{s(u,t)} > \bar{m}_s$ gives $e_{s(u,t)} > 1$. With this—and $m_{s(u,t)} \geq 3$—we specify: if $e_{s(u,t)} < 1$, then engagement is low and $f_{u,t} = \texttt{LE}$, whereas if $e_{s(u,t)} \geq 1$ it is high and $f_{u,t} = \texttt{HE}$.

Course content embeddings. We now detail our approach to processing course content data into features. As discussed, due to the limited textual information in this application, applying standard natural language processing methods (such as word count techniques [17] and LDA) may not be sufficient. Instead, we resort to statistical language models that are pre-trained on web-scale data; in particular, we use GloVe embeddings [26], a word-to-vector mapping pre-trained on the Wikipedia 2014 and Gigaword 5 datasets. These embeddings are well suited as inputs to RNNs, since the Euclidean distance (or cosine similarity) between GloVe vectors provide useful insights into the linguistic similarities between the corresponding words [26].

Specifically, we seek a vector representation $\mathbf{y}_s$ for segment s that quantifies the material covered in s based on the bag-of-words $\mathbf{x}_s$. We first map each word $w_k \in \mathscr{X}$ to its corresponding vector $\mathbf{y}_k \in \mathbb{R}^{100}$ in the pre-trained GloVe library,[3] where 100 is the choice of dimension in the pre-trained embedding. We then aggregate the word vectors in s to obtain the embedding $\mathbf{y}_s = \sum_k \mathbf{x}_s(k) \cdot \mathbf{y}_k \in \mathbb{R}^{100}$. To reduce the number of parameters under consideration, we further perform dimensionality reduction of $\mathbf{y}_s$ via principal component analysis (PCA) [15], obtaining $\mathbf{y}_s \in \mathbb{R}^D$ for a parameter D; $\mathbf{y}_s$ is taken as the top-D principal components of the PCA. We will consider the choice of D in our experiments section.

3. RNN-BASED MODEL

We now propose an RNN-based probabilistic model for learner behavior that uses the features defined in Section 2. The reason that we choose RNN as a basis is that it is often used to model sequential data, such as text [13, 16] and user purchasing activities [25], which is characteristic of learner behavioral sequences as well.

Our overall model architecture is visualized in Figure 2. It consists of two main parts described in this section: (i) a latent knowledge state transition model, and (ii) a learner behavior model.

3.1 Latent Knowledge State Transition Model

The state transition model is similar to that of generic RNNs. In our context, the transition is induced by gaining knowledge from watching a video segment. Letting $\mathbf{h}_{u,t} \in \mathbb{R}^K$ denote the K-dimensional knowledge state vector of learner u at time t, we model the transition as

$$\mathbf{h}_{u,t} = \sigma\left(\mathbf{W}\mathbf{h}_{u,t-1} + \mathbf{l}_{u,t-1} + \mathbf{b}\right), \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{K \times K}$ denotes the state transition parameter matrix and $\mathbf{b} \in \mathbb{R}^K$ denotes the bias vector. $\sigma(\cdot)$ is a nonlinear function, for which we will test a range of possible nonlinearities later in the experiments section. $\mathbf{l}_{u,t-1}$ is defined as

$$\mathbf{l}_{u,t-1} = e_{u,t-1}\mathbf{U}\mathbf{y}_{s(u,t-1)}$$

to quantify the amount of knowledge the learner acquires from watching segment $s(u,t-1)$ (a setting that follows [21]); $\mathbf{U}\mathbf{y}_{s(u,t-1)}$ captures the knowledge contained in this segment, since $\mathbf{y}_{s(u,t-1)}$ is its GloVe embedding and $\mathbf{U} \in \mathbb{R}^{K \times D}$ is the input parameter matrix that maps its text embedding to latent knowledge, while $e_{u,t-1}$ is the scalar engagement variable which dictates the amount of $\mathbf{U}\mathbf{y}_{s(u,t-1)}$ transferred to the learner. We parameterize $e_{u,t}$ with the behavioral feature $f_{u,t}$:

$$e_{u,t} = \begin{cases} e_h & \text{if } f_{u,t} = \texttt{HE} \\ e_l & \text{if } f_{u,t} = \texttt{LE} \\ 0 & \text{if } f_{u,t} \in \{\texttt{SB}, \texttt{SF}\}. \end{cases}$$

Here, $e_h, e_l \in [0,1]$ are parameters that characterize specific engagement levels that HE and LE correspond to in our model. If a learner skips a video segment, their engagement level is zero, so no knowledge is gained.

[2] For the 20 sec video segments in this course, we set $\bar{m}_s = 20$ and $\alpha = 0.1$ by default.

[3] https://nlp.stanford.edu/projects/glove/

Note that this characterization of engagement differs from that described in [8, 20]. In our model, when there is no knowledge input ($\mathbf{l}_{u,t-1} = \mathbf{0}$), $\mathbf{W}$ and $\mathbf{b}$ can be used to characterize other causes of knowledge state transition, e.g., forgetting. For another example on the relationship between engagement and learning, see [29].

3.2 Learner behavior model

The behavior model concerns the feature variable $f_{u,t}$. We model the probability that a learner selects each $f \in \mathscr{F}$ with the following softmax distribution:

$$P(f_{u,t} = f) = \frac{e^{\mathbf{v}_f^T[\mathbf{g}_{u,t}^T\ \mathbf{z}_{u,t}^T]^T + d_f}}{\sum_{f' \in \mathscr{F}} e^{\mathbf{v}_{f'}^T[\mathbf{g}_{u,t}^T\ \mathbf{z}_{u,t}^T]^T + d_{f'}}}, \qquad (2)$$

where the variables are $\mathbf{g}_{u,t} \in \mathbb{R}^K$, $\mathbf{z}_{u,t} \in \mathbb{R}^K$, $\mathbf{v}_f \in \mathbb{R}^{2K}$, and $d_f \in \mathbb{R}$. The vectors $\mathbf{v}_f$ and the biases d_f, together with latent state variables $\mathbf{g}_{u,t}$ and $\mathbf{z}_{u,t}$, decide learner behaviors on each video segment. $\mathbf{g}_{u,t}$ denotes the prerequisite knowledge gap and $\mathbf{z}_{u,t}$ denotes the learning goal knowledge gap; they are defined from the knowledge state transition model as follows:

Prerequisite knowledge gap: $\mathbf{g}_{u,t} := \mathbf{p}_{s(u,t)} - \mathbf{r}_{u,t}$ is the prerequisite knowledge gap vector. $\mathbf{p}_s$ denotes the required knowledge level of segment s, and $\mathbf{r}_{u,t}$ denotes the portion of learner u's knowledge state at time t that is relevant to the prerequisite requirement of segment $s(u,t)$. Concretely, $\mathbf{r}_{u,t}$ is defined as

$$\mathbf{r}_{u,t} = \sum_{\tau=1}^{t-1} \mathbf{R}_{s(u,\tau),s(u,t)} \cdot \mathbf{l}_{u,\tau},$$

where the matrix $\mathbf{R} \in \{\mathbb{R}_+ \cup 0\}^{S\times S}$, at the core of our model, characterizes the prerequisite structure among segments. A large value of $\mathbf{R}_{s,s'}$ implies segment s is a strong prerequisite of s', while $\mathbf{R}_{s,s'} = 0$ means s is not a prerequisite of s'. Note that the nonnegativity constraint placed on the prerequisite structure matrix is necessary for interpretability of the model parameters, since reversing the sign of every parameter would lead to the same data likelihood, rendering the model unidentifiable in the absence of this constraint.

Learning goal knowledge gap: $\mathbf{z}_{u,t} := \mathbf{c}_u - \mathbf{h}_{u,t-1}$ denotes the learning goal knowledge gap vector. $\mathbf{c}_u$ characterizes the learning goal of learner u, i.e., a target knowledge state that they are satisfied upon reaching, while $\mathbf{h}_{u,t-1}$ denotes their previous knowledge state. In general, $\mathbf{c}_u$ can either be personally imposed (e.g., in optional, recreational learning) or externally enforced (e.g., in institutionalized learning); for the course in this paper, it is the latter.

Model intuition. Our model is based on the intuition that there are two factors driving a learner's behavior while watching a particular video segment. The setup of these two factors enables us to extract the prerequisite dependencies ($\mathbf{R}$) among video segments by observing the sequences of learner behaviors.

The first factor, parameterized by the prerequisite knowledge gap vector $\mathbf{g}_{u,t}$, characterizes whether the learner possesses enough prerequisite knowledge to master the current segment. This gap is given by the difference between the knowledge level required to master the current segment ($\mathbf{p}_{s(u,t)}$) and the learner's accumulated knowledge from prerequisite segments ($\mathbf{r}_{u,t}$). The learner would have gained such knowledge by exhibiting high engagement ($f_{u,t} = \texttt{HE}$) on the prerequisite segments; if they do not have enough, they are more likely to skip backwards ($f_{u,t} = \texttt{SB}$) to study further.

The second factor, parameterized by the learning goal knowledge gap vector $\mathbf{z}_{u,t}$, characterizes whether the learner has already reached their learning goal. This gap is given by the difference between the goal ($\mathbf{c}_u$) and the learner's previous knowledge state ($\mathbf{h}_{u,t-1}$). If the learner has already accumulated enough knowledge, they are more likely to exhibit low engagement ($f_{u,t} = \texttt{LE}$) or to skip forward ($f_{u,t} = \texttt{SF}$).

Parameter inference. We estimate the latent model parameters, i.e., the input, transition, and output parameters $\mathbf{U}$, $\mathbf{W}$, and $\mathbf{v}_f$, the biases $\mathbf{b}$, the latent engagement level parameters e_h and e_l, the learning goal vectors $\mathbf{c}_u$, and the prerequisite structure matrix $\mathbf{R}$ by using the Adagrad optimizer [11] to minimize the cross-entropy loss [12] on the observed behavior sequences. The cross-entropy loss is the standard loss function for categorical data (each category corresponds to a behavior in $\mathscr{F} = \{\texttt{LE}, \texttt{HE}, \texttt{SB}, \texttt{SF}\}$). We implement our inference algorithm in TensorFlow.[4]

4. EXPERIMENTS

In this section, we evaluate our model proposed in Section 3 on the product development course. We first describe our experimental setup, including training/validation and tuning procedures. Then, we investigate the ability of our model to predict learner behavior on future video segments, compared to baselines. Once we have established model quality, we perform an exploratory analysis of the prerequisite structure information in the model, and present the results from sharing these insights with a course administrator.

4.1 Experimental Setup

Training and validation. We partition the original dataset to two parts: (i) the training set, which is used to train models, and (ii) the validation set, which is used to evaluate prediction performance. We randomly select 90% of the learners to form the training set and use the remaining 10% as the test set.

In each training epoch, we randomly select 800 learners from the training set and use their behavioral data to calculate the gradient of the overall cross-entropy loss with respect to our model parameters. We then take a gradient step using the Adagrad optimizer [11] and evaluate the prediction performance of our model on the validation set. Note that since the learners in the validation set are not used in our training procedure, we do not have estimates of their target knowledge state vector $\mathbf{c}_u$. Therefore, we take the average of the estimated target knowledge state vectors over learners in the training set and use it for learners in the validation set.

Metrics. We report the performance of our proposed model and baselines using two standard evaluation metrics on the validation dataset: (i) the cross entropy loss, and (ii) prediction accuracy, which is simply the percent of behaviors that are predicted correctly. Lower loss and higher accuracy implies better performance.

Baselines. We focus on shallow RNN-type networks as baselines, since (i) they have been widely used to model sequential data

[4] https://www.tensorflow.org/

and (ii) they have a similar architecture to our model, thereby providing a fair comparison.

First, we consider an RNN model with content GloVe embeddings $\mathbf{y}_s$ as input and learner behaviors $f_{u,t}$ as output, which we refer to as *RNN-G*:

$$\mathbf{h}_{u,t} = \sigma(\mathbf{U}\mathbf{y}_{s(u,t)} + \mathbf{W}\mathbf{h}_{u,t-1} + \mathbf{b})$$
$$P(f_{u,t} = f) = \frac{e^{\mathbf{v}_f^T \mathbf{h}_{u,t} + b_f}}{\sum_{f' \in \mathscr{F}} e^{\mathbf{v}_{f'}^T \mathbf{h}_{u,t} + b_{f'}}}.$$

In RNN-G, the input at every time step does not contain the learner's actual behavior in the last time step. Such a setting can be disadvantageous when the input provides only limited information on the current output. To investigate this, we also consider an RNN model that feeds the ground truth behavior from the last time step ($f_{u,t-1}$) back into the model as input at the current time step, which we refer to as *RNN-F*:

$$\mathbf{h}_{u,t} = \sigma(\mathbf{U}\mathbf{f}_{u,t-1} + \mathbf{W}\mathbf{h}_{u,t-1} + \mathbf{b})$$
$$P(f_{u,t} = f) = \frac{e^{\mathbf{v}_f^T \mathbf{h}_{u,t} + b_f}}{\sum_{f' \in \mathscr{F}} e^{\mathbf{v}_{f'}^T \mathbf{h}_{u,t} + b_{f'}}}.$$

Here, we slightly abuse notation, using $\mathbf{f}_{u,t-1} \in \{0,1\}^{|\mathscr{F}|}$ to denote the one-hot-encoded vector version of the observed learner action at time $t-1$ [12]. Note that this network structure has been used to model sequential data, e.g., text; this technique is sometimes referred to as teacher forcing [36].

These two baselines—RNN-G and RNN-F—can both use information from previous time steps for the prediction of learner behavior at the current time step. In some sequential prediction tasks, only recent information is needed, whereas in other scenarios, long-term dependencies must be considered; the latter may especially be true in learning given how material builds on itself [38]. Since neither RNN-G nor RNN-F support the use of information from several time steps back, we will also consider the long short-term memory (LSTM) network as a baseline algorithm, which we will refer to as *LSTM*. Similar to RNN-F, we use previous learner behavior as the input to the next time step in LSTM. The comparison between our model and LSTM will show which is better at storing and retrieving information from further back in time.

Parameter tuning. Several parameters must be tuned to optimize the performance of each model. First is the dimension of the latent knowledge state vector K, which applies to all models: we sweep over $K \in \{5, 10, \ldots, 55\}$. Second is the dimension of the GloVe embedding D, for our model and RNN-G: we consider $D \in \{5, 10, \ldots, 45\}$, where D corresponds to the top-D principal components of the PCA on the segment vectors.

We also examine the performance of our model with different choices of the nonlinearity function $\sigma(\cdot)$. For this, we use the nonlinearities built in to TensorFlow: rectified linear units (relu), exponential linear units (elu), hyperbolic tangent (tanh), soft plus (softplus), and no nonlinearity (identity).

Through our experiments, we found that a constant learning rate of 0.01 and a total of 300-500 training epochs consistently led to the best results, for all three baseline algorithms. As a result, we will not perform more than 350 training epochs, since the performance does not significantly improve after that.

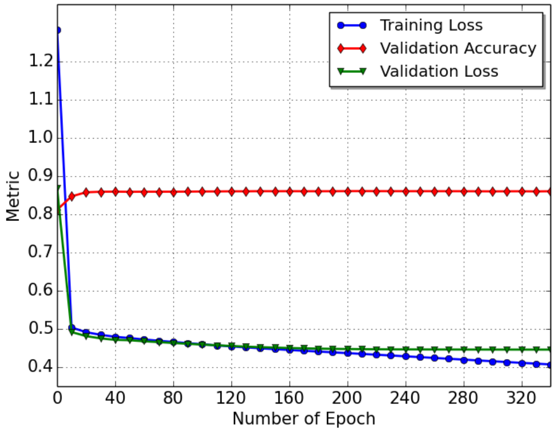


Figure 3: Performance of our model against the number of training epochs. While the training loss continues to decrease, the validation loss stabilizes quickly after approximately 200 epochs.

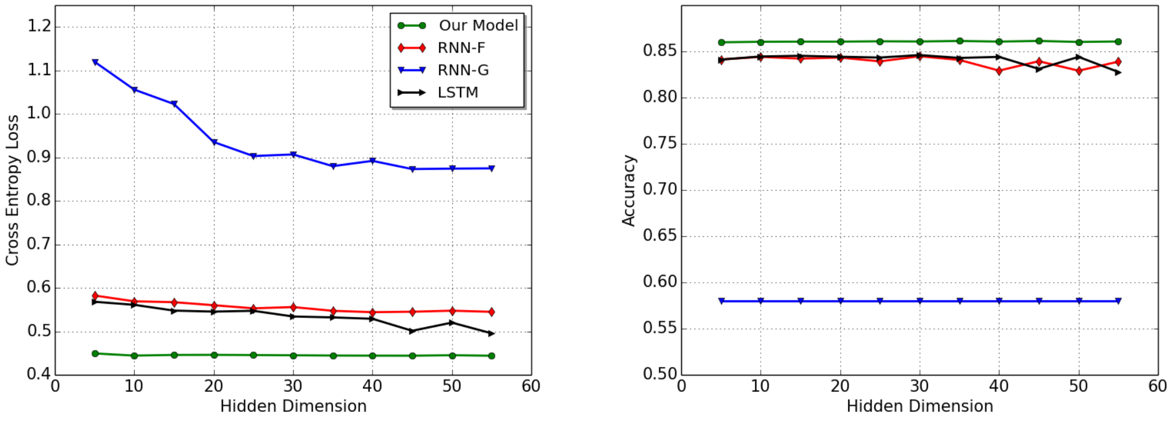
4.2 Prediction Performance

We consider model performance against several parameters. When parameters are constant, they take the default values of $K = 45$, $D = 30$, and $\sigma = \tanh$.

Varying number of training epochs. In Figure 3, we plot the cross entropy loss on both the training and validation sets, as well as the accuracy on the validation set, as the number of training epochs is varied for our model. We see that (i) the training loss exhibits a continually decreasing trend with minimal fluctuations, while (ii) the validation loss drops quickly initially but stabilizes after around 200 epochs, and (iii) the validation accuracy stabilizes quickly after about 20 epochs. Since the performance on the validation set remains stable after a large number of epochs, we conclude that *our model does not easily overfit*. In fact, implementing dropout regularization [30] showed minimal impact on the performance of our model. Therefore, we did not use dropout or any other form of regularization in our other experiments.

Varying latent knowledge state dimension K. In Figure 4, we plot (a) the cross entropy loss and (b) the accuracy of all four models on the validation set against the dimension of the hidden layer K. Overall, we see that *our model outperforms every baseline for each choice of K, and significantly so on the cross entropy loss metric*, which demonstrates the ability of our model to accurately predict learner behavior. While all models show improving performance as K increases, after $K = 10$ the improvement for our model is minimal. The fact that both our model uses the same input information yet outperforms RNN-F justifies our particular design choices involving the prerequisite knowledge gap and learning goal knowledge gap vectors.

We also see that RNN-G performs significantly worse than RNN-F. This observation suggests that *the features given by the content data provide only limited information on learner behavior*, which validates our conjecture that the learning content itself (i.e., the video transcripts) is a very limited data source. Finally, we note



(a) Cross entropy loss

(b) Accuracy

Figure 4: Prediction performance on the validation set as the dimension of the latent knowledge state vector (K) is varied. Our model outperforms all baselines in each case tested, especially on the cross entropy loss metric, indicating an overall ability to predict learner behavior. Moreover, the performance of our model is robust to the choice of K.

that among the baselines, LSTM slightly outperforms RNN-F, indicating that *in our application of online learning, there is benefit to preserving information on behavior further back in time*.

Varying input dimension D. In Figure 5, we plot (a) the cross entropy loss and (b) the accuracy of our proposed model and RNN-G against the dimension of the input GloVe embedding D on the validation set. Overall, we see that *the performance of both models is insensitive to the choice of D*. One possible explanation is that even with very low-dimensional input (i.e., taking only the top few principal components), the embeddings still encapsulate the video transcript text effectively. To investigate this, in Figure 5, we label the percentage of variance explained by the top-D principal components of the GloVe embedding for every value of D. We see that the top-5 principal components (i.e., $D = 5$) explain about 95% of the total variance, which explains why increasing D beyond $D = 5$ does not further improve the performance. This observation on the percentage of variance explained provides more evidence that the information contained in the textual content is limited.

Varying nonlinearity σ. In Table 1, we tabulate the cross entropy loss and accuracy of our model on the validation set using the different non-linearity functions σ. Overall, while the elu nonlinearity achieves the best performance when considering both metrics, every choice of nonlinearity leads to very similar performance. This suggests that *our model is robust to the choice of nonlinearity in the latent knowledge state transition*.

4.3 Prerequisite Structure Analysis

Having established overall model quality, we now analyze the extracted prerequisite structure, i.e., the model matrix $\mathbf{R}$. In doing so, we will consider several examples that illustrate how the course was constructed, referring to the video titles and segment transcripts as needed. We then validate the insights through the results of a questionnaire on some of the particular findings that was provided to a course administrator. This administrator possesses intimate knowledge of the course content and how it was constructed.

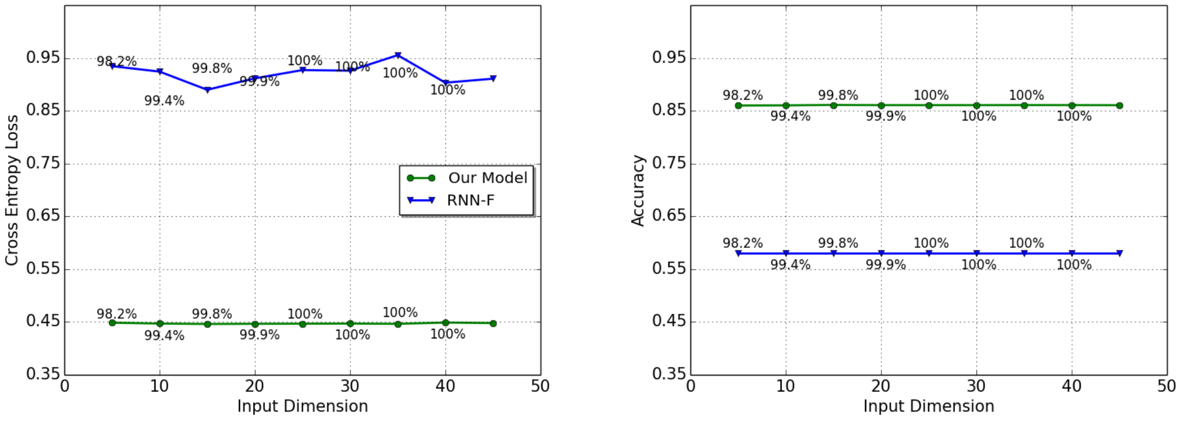
To derive the insights, we consider two different cases of the matrix: (a) $\mathbf{R}$ across the entire course, obtained from extracting the prerequisite structure between all video segments, and (b) $\mathbf{R}^v$ for each video v, from estimating the structure between segments in each video separately. Case (a) uses the results for $K = 45$, $D = 30$, $\sigma = \tanh$ from the previous experiment, while case (b) is a new experiment with these settings.

4.3.1 Insights: Full course matrix

Figure 6(a) visualizes $\mathbf{R}$ across the course. We focus on a few key findings here, some across videos and some for individual segments. First is that *segments in the last two videos have substantially more prerequisites than those in the first two*. The only segment with significant prerequisites in the first two is Segment 8, while the only one without significant prerequisites in the second two is Segment 13. At a high level, then, we can infer that the first two videos are laying the groundwork for material covered later on. This makes sense considering even just the titles of the videos, with the first two geared towards explaining the "vision" and reasoning for the development of this product, and the later two expounding on the product's "features" and technical description.[5]

For individual segments, consider Segment 8 from the previous discussion. This segment *has all previous ones as prerequisites, with some more significant than others*. The transcript for this segment indicates a discussion on the demand for this type of product over the next several years, which is traditionally viewed as "problematic," so it makes sense that learners should study Segments 0 to 7 first to understand the "vision" of this version of the product to mitigate the problem. Segments 1, 4, and 6 discuss "problem mitigation" in particular, consistent with them being larger prerequisites. Another interesting case is Segment 26, for which *there are several prerequisite segments throughout the course, but the one immediately previous is not as significant*. Segment 26 actually continues with the theme of "problem mitigation," which is discussed in Segment 24 but not in Segment 25. Segments 4, 11, and 19 reference the particular method of "problem mitigation," which is also con-

[5] We omit exact video titles and transcripts in this section to preserve anonymity, but provide enough context for the key points.



(a) Cross entropy loss.

(b) Prediction accuracy.

Figure 5: Prediction performance on the validation set as the dimension of the input word embedding (D) is varied for both our model and RNN-F. For each point, we label the percentage of variance in the input explained by the top-D principal components. The performance remains largely unchanged as D increases in each case, which is consistent with over 98% of the variance being explained by the top-5 principal components (i.e., $D = 5$).

Activation Functions	Formula	Accuracy	Cross Entropy Loss
relu	$\sigma(x) = x$ if x ≥ 0, $\sigma(x) = 0$ if $x < 0$	**0.861**	0.444
tanh	$\sigma(x) = \frac{1-e^{-x}}{1+e^{-x}}$	**0.861**	0.445
elu	$\sigma(x) = x$ if x ≥ 0, $\sigma(x) = e^x - 1$ if $x < 0$	**0.861**	**0.443**
softplus	$\sigma(x) = ln(1+e^{-x})$	**0.861**	0.447
identity	$\sigma(x) = x$	0.860	0.454

Table 1: Performance of our model with different choices of nonlinearity $\sigma(\cdot)$. Except for the identity (no nonlinearity) which performs worse, all nonlinearities lead to a similar performance, implying that our model is robust to the choice of nonlinearity.

sistent with them being strong prerequisites to Segment 26.

4.3.2 Insights: Individual video matrices

Figure 6(b) visualizes $\mathbf{R}^v$ for separate videos v. Compared with Figure 6(a), it is easier to compare segments within videos, but the relative magnitudes of prerequisites between videos is lost. For Video 4, we see that *prerequisites within the video tend to become weaker as the video progresses*, which is not obvious in Figure 6(a). For example, while Segment 23 has a heavy dependence on Segment 22, Segment 34 is only lightly dependent on a few segments in the video. Being close to the end, Segment 34 is summarizing information across the course, which is evident through its prerequisites in Figure 6(a). The inferred relation between Segments 23 and 24 is consistent with both of these segments' transcripts discussing particular technologies in the new product.

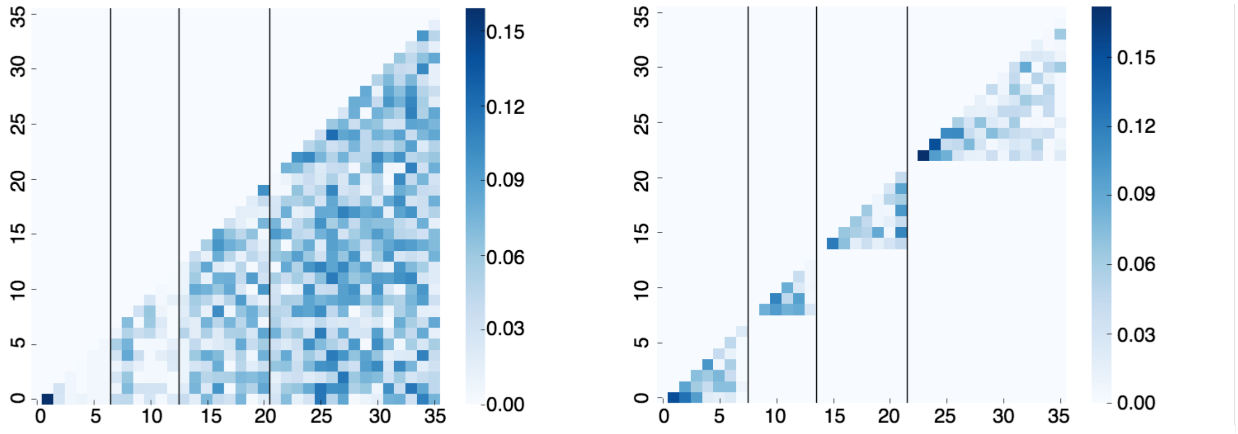
Another insight is that *with the exception of Video 3, the last segment in each video has only light prerequisites within the video*. Intuitively, we would expect last segments to summarize the material covered in the video, but such a review may not constitute a strong prerequisite. The transcript of Video 3's concluding Segment 21, on the other hand, indicates that it is a continuation of the "product features" discussion.

4.3.3 Questionnaire and response

The questionnaire provided to the course administrator began with a brief description of the algorithm and purpose. It then included a visualization of the $\mathbf{R}$ matrix, and an enumeration of several statements drawn from our insights ranging from conclusions on particular segments to general trends across multiple segments. A sample statement provided is "this segment does not have any prerequisites, i.e., studying prior segments is not helpful to its understanding." The task of the course administrator was to indicate their level of agreement with each statement on a five-point Likert Scale, from 1 (strong disagreement) to 5 (strong agreement).

80% of the responses we obtained to the statements were in the range of 4-5. This indicates that the course administrator generally agreed with the the prerequisite dependencies extracted by our algorithm, and in turn gives additional validity to our proposed model in terms of its ability to generate human-interpretable insights.

The disagreements tended to be for statements that compared the magnitude to which two particular segments were prerequisites to another segment, i.e., claiming that one was a stronger prerequisite to the segment than the other. Since the agreements, by contrast, were on more general statements concerning the existence and/or strength of prerequisites to a given segment or group of segments (e.g., "segment 1 is a strong prerequisite to segment 2", "segments in part 1 of the course tend to have more dependencies than segments in part 3"), our algorithm may not differentiate magnitudes of prerequisites for a particular segment well. There are several possible reasons for this. One is the method used to segment the content: rather than choosing uniform 20 second chunks of video, for example, it may be desirable to incorporate segmentation into the modeling procedure, e.g., by maximizing the difference in prerequisites between adjacent segments. Another is the treatment and



(a) Full course **R** (b) Individual videos $\mathbf{R}^v$

Figure 6: Visualizations of the prerequisite matrices extracted in two ways: (a) **R** across the entire course, and (b) $\mathbf{R}^v$ for each video v separately. The (s,s')th entry (the entry on the sth row and s'th column, with $s < s'$) characterizes how much segment s serves as a prerequisite of segment s'. The solid lines delineate the four different videos.

presentation of the values comprising the **R** matrix: rather than reporting these as real numbers, it may be desirable to group them into relative magnitudes, e.g., low/medium/high or a simple binary indicator of whether there is a noteworthy dependency. Educators may be more interested in broader distinctions.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a recurrent neural network-based model to extract prerequisite structure among fine-granular pieces of learning content. We modeled such prerequisite structure information as latent variables, and extracted it from learner behavioral data. We applied our model to an online course dataset that contains the clickstream activity behavioral data from 12,000 learners watching course videos. Our experiments showed that our model significantly outperforms baseline models in predicting learner behavior and, more importantly, that it effectively extracts both intra- and inter-video prerequisite dependencies among video segments; we were able to verify these insights through responses to a questionnaire provided to a course administrator. More generally, our work demonstrated that large-scale learner behavioral data can offer interesting insight into learning content; therefore, it is important to use learner behavioral data to aid content analytics, especially when content data is sparse and learner performance data is unavailable.

There are several avenues of future work. One is experimentally testing whether the extracted prerequisite structure can lead to better personalized remediation or enrichment activities selection [5, 19, 27]. Another is adapting our model to other content types, e.g., educational games [23]. Also, one can try to adapt out model to extract prerequisite structures in longer (e.g., semester-long) courses by aggregating learner behavior at a higher granularity level, and compare the results against that obtained via traditional, content data-based methods. Moreover, to further improve the insights provided by our model, two approaches can be investigated as discussed: incorporating segmentation into the model itself to e.g., maximize the difference in prerequisites between adjacent segments, and grouping the values in the **R** matrix into discrete categories. Finally, additional slack variables can be incorporated into to our model to allow variation in learner behaviors; learners sometimes make poor assessments about their prerequisite knowledge and are unable to navigate across the course efficiently.

In particular, for personalization, note that the prerequisite structures (the **R** matrix) our model extracts can drive automated content individualization. For example, when learner u reaches segment s' at time t, a course delivery system could check whether the prerequisite knowledge gap $\mathbf{g}_{u,t} \geq \mathbf{0}$. If not, then a combination of segments s for which $\mathbf{R}_{s,s'}$ is high and engagement e_s is low (i.e., significant prerequisites that the learner has not studied) can be displayed first. The system could then update $\mathbf{g}_{u,t}$ as these prerequisites are studied, and "unlock" the segment s' once the learner has engaged with them enough (when the prerequisite knowledge gap $\mathbf{g}_{u,t}$ diminishes). We are currently implementing such a method.

6. REFERENCES

[1] R. Baker and P. Inventado. *Educational Data Mining and Learning Analytics*, pages 61–75. Springer, 2014.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Drichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, Jan. 2003.

[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proc. ACM SIGMOD International Conference on Management of Data*, pages 1247–1250, June 2008.

[4] A. Botelho, H. Wan, and N. Heffernan. The prediction of student first response using prerequisite skills. In *Proc. ACM Conference on Learning at Scale*, pages 39–45, Mar. 2015.

[5] C. Brinton, R. Rill, S. Ha, M. Chiang, R. Smith, and W. Ju. Individualization for education at scale: MIIC design and preliminary evaluation. *IEEE Transaction on Learning Technology*, 8(1):136–148, Jan. 2015.

[6] E. Brunskill. Estimating prerequisite structure from noisy data. In *Proc. International Conference on Educational Data Mining*, pages 217–222, July 2011.

[7] D. Chaplot, Y. Yang, J. Carbonell, and K. Koedinger. Data-driven automated induction of prerequisite structure graphs. In *Proc. International Conference on Educational*

Data Mining, pages 318–323, July 2017.

[8] W. Chen, C. G. Brinton, D. Cao, and M. Chiang. Behavior in Social Learning Networks: Early Detection for Online Short-Courses. In *Proc. IEEE International Conference on Computer Communnications*, 2017.

[9] Y. Chen, J. González-Brenes, and J. Tian. Joint discovery of skill prerequisite graphs and student models. In *Proc. International Conference on Educational Data Mining*, pages 46–53, July 2014.

[10] Y. Chen, P. Wuillemin, and J. Labat. Discovering prerequisite structure of skills through probabilistic association rules mining. In *Proc. International Conference on Educational Data Mining*, pages 117–124, June 2015.

[11] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Machine Learning Research*, 12:2121–2159, July 2011.

[12] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[13] A. Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv: 1308.0850*, June 2013.

[14] S. Han, J. Yoon, and J. Yoo. Discovering skill prerequisite structure through Bayesian estimation and nested model comparison. In *Proc. International Conference on Educational Data Mining*, pages 398–399, June 2017.

[15] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2010.

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, Nov. 1997.

[17] K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[18] I. Labutov, Y. Huang, P. Brusilovsky, and D. He. Semi-supervised techniques for mining learning outcomes and prerequisites. In *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 907–915, Aug. 2017.

[19] A. S. Lan and R. G. Baraniuk. A contextual bandits framework for personalized learning action selection. In *Proc. International Conference on Educational Data Mining*, pages 424–429, June 2016.

[20] A. S. Lan, C. G. Brinton, T. Yang, and M. Chiang. Behavior-based latent variable model for learner engagement. In *Proc. International Conference on Educational Data Mining*, pages 64–71, June 2017.

[21] A. S. Lan, C. Studer, and R. G. Baraniuk. Time-varying learning and content analytics via sparse factor analysis. In *Proc. 20th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pages 452–461, Aug. 2014.

[22] C. Liang, Z. Wu, W. Huang, and C. Giles. Measuring prerequisite relations among concepts. In *Proc. Conference on Empirical Methods in Natural Language Processing*, pages 1668–1674, Sep. 2015.

[23] Y. Liu, T. Mandel, E. Brunskill, and Z. Popovic. Trading off scientific knowledge and user learning with multi-armed bandits. In *Proc. 7th Intl. Conf. Educ. Data Min.*, pages 161–168, July 2014.

[24] I. Manickam, A. S. Lan, and R. G. Baraniuk. Contextual multi-armed bandit algorithms for personalized learning action selection. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6344–6348, Mar. 2017.

[25] M. Mozer, R. Lindsey, and D. Kazakov. Neural Hawkes process memories. In *Proc. NIPS Symposium on Recurrent Neural Networks*, Dec. 2016.

[26] J. Pennington, R. Socher, and C. D. Manning. GloVe: Global vectors for word representation. In *Proc. Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, Oct. 2014.

[27] S. Reddy, I. Labutov, and T. Joachims. Learning student and content embeddings for personalized lesson sequence recommendation. In *Proc. ACM Conference on Learning at Scale*, pages 93–96, Apr. 2016.

[28] R. Scheines, E. Silver, and I. Goldin. Discovering prerequisite relationships among knowledge components. In *Proc. International Conference on Educational Data Mining*, pages 355–356, July 2014.

[29] S. Slater, R. Baker, J. Ocumpaugh, P. Inventado, P. Scupelli, and N. Heffernan. Semantic features of Math problems: Relationships to student learning and engagement. In *Proc. International Conference on Educational Data Mining*, pages 223–230, June 2016.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Machine Learning Research*, 12:1929–1958, June 2014.

[31] P. Talukdar and W. Cohen. Crowdsourced comprehension: Predicting prerequisite structure in wikipedia. In *Proc. Workshop on Building Educational Applications Using NLP*, pages 307–315, June 2012.

[32] A. Vuong, T. Nixon, and B. Towle. Estimating prerequisite structure from noisy data. In *Proc. International Conference on Educational Data Mining*, pages 211–216, July 2011.

[33] H. Wan and J. Beck. Considering the influence of prerequisite performance on wheel spinning. In *Proc. International Conference on Educational Data Mining*, pages 129–135, June 2015.

[34] S. Wang, C. Liang, Z. Wu, K. Williams, B. Pursel, B. Brautigam, S. Saul, H. Williams, K. Bowen, and C. Giles. Concept hierarchy extraction from textbooks. In *Proc. ACM Symposium on Document Engineering*, pages 147–156, Sep. 2015.

[35] S. Wang, A. Ororbia, Z. Wu, K. Williams, C. Liang, B. Pursel, and C. Giles. Using prerequisites to extract concept maps from textbooks. In *Proc. ACM International on Conference on Information and Knowledge Management*, pages 317–326, Oct. 2016.

[36] R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, 1989.

[37] X. Xiong, S. Adjei, and N. Heffernan. Improving retention performance prediction with prerequisite skill features. In *Proc. International Conference on Educational Data Mining*, pages 375–376, July 2014.

[38] T. Yang, C. G. Brinton, C. Joe-Wong, and M. Chiang. Behavior-based grade prediction for MOOCs via time series neural networks. *IEEE J. Selected Topics in Signal Processing*, May 2017.

Principles for Assessing Adaptive Online Courses

Weiyu Chen
Zoomi Inc.
weiyu.chen@zoomiinc.com

Carlee Joe-Wong
Carnegie Mellon University
carlee.joe-wong@west.cmu.edu

Christopher G. Brinton
Zoomi Inc.
chris.brinton@zoomiinc.com

Liang Zheng
Princeton University
liangz@princeton.edu

Da Cao
Zoomi Inc.
da.cao@zoomiinc.com

ABSTRACT

Adaptive online courses are designed to automatically customize material for different users, typically based on data captured during the course. Assessing the quality of these adaptive courses, however, can be difficult. Traditional assessment methods for (machine) learning algorithms, such as comparison against a ground truth, are often unavailable due to education's unique goal of affecting both internal user knowledge, which cannot be directly measured, as well as external, measurable performance. Traditional metrics for education like quiz scores, on the other hand, do not necessarily capture the adaptive course's ability to present the right material to different users. In this work, we present a mathematical framework for developing scalable, efficiently computable metrics for these courses that can be used by instructors to gauge the efficacy of the adaptation and their course content. Our metric framework takes as input a set of quantities describing user activities in the course, and balances definitions of user consistency and overall efficacy as inferred by the quantity distributions. We support the metric definitions by comparing the results of a comprehensive statistical analysis with a sample metric evaluation on a dataset of roughly 5,000 users from an online chess platform. In doing so, we find that our metrics yield important insights about the course that are embedded in the larger statistical analysis, as well as additional insights into student drop-off rates.

1. INTRODUCTION

Online learning has become a popular way for universities, corporations, and other institutions to offer full classes and certification programs at scale to students outside the traditional campus setting. Yet students in these courses, particularly in those with open enrollment such as Massive Open Online Courses (MOOCs), often exhibit a wide range of backgrounds, degrees of preparedness, and goals. For example, while some may wish to indulge a personal interest, others may wish to refresh their memory of the course material in preparation for a job [11].

Adaptive online courses automatically individualize the content presented to users, and thus hold promise of accommodating student heterogeneity at scale [4]. These course delivery systems may leverage a wide array of measurements to personalize material, such as user performance on assessments and user behavior exhibited while interacting with content and in discussion forums [4]. Both of these forms of data – behavioral and performance – have been shown to be predictive of learning outcomes [2, 6], indicating that they contain information about whether a user's goals have been met. Fully analyzing the different types of user behavior and performance in a course, however, may prove to be overwhelming to an instructor, and may require significant knowledge of statistics in order to properly interpret the analysis.

Thus, it is useful to develop summary metrics that break down insights from user data into a few easily understandable statistics, particularly for large-scale online courses. Such metrics may also allow direct comparisons of the effectiveness of different courses, or of different units within a course. In this work, we propose a mathematical framework and guidelines for such metrics, and demonstrate particular versions of them on a MOOC dataset.

1.1 Research Challenges and Metric Requirements

Education influences both (i) externally observable activity during a course (*e.g.,* performance on quizzes) and (ii) internal user states during and after a course (*e.g.,* knowledge transfer from the course to the workplace) [12]. Any metrics for online (or offline) course efficacy should account for changes in both, but internal changes cannot be observed directly and are often approximated by responses to quiz questions, which are themselves external. For this reason, it is nearly impossible to define a single "ground truth" measure of course quality through conventional learning measurements [24]. Online courses can compensate for this difficulty by collecting many different types of user data, including both user performance as well as user behavioral measurements, which can give a rich picture of how users benefit from the content. At the same time, integrating insights from heterogeneous sources of learning data is itself a challenging task [2].

Adaptive online courses add a further challenge beyond heterogeneous data: unlike non-adaptive courses, they are designed to offer users a consistent experience. A course evaluation criterion must then account for not only overall course *efficacy*, but also its *consistency* across users: such consistency encapsulates how well the adaptation can account for different users and helps to ensure robustness to new, possibly different users joining the system [9]. We therefore identify the following three research challenges:

C1. Incorporating heterogeneous user data: There are at least three types of user data: (i) *behavioral*, *e.g.,* clickstream measurements on course content, (ii) *performance*, both within and external to the course, and (iii) *navigation*, measuring how closely users follow their adaptation path. A metric should be able to combine all or only a subset of this data, and/or other sources, depending on what data is available.

Each of these three measurement types can provide different insights into course efficacy. For instance, some users may obtain high quiz performance while spending a minimal amount of time engaging with the content. This would indicate "success" if a user simply wished to master the course material, but "failure" if he/she also wanted to be intellectually challenged [4]. The navigation data could shed light on this distinction: those who deviate from the recommended path are probably searching for additional material, while those following it are satisfied with the content provided [2]. By combining different types of user measurements, a metric can account for the fact that a low score in one type may not necessarily indicate an ineffective course.

C2. Balancing user consistency with efficacy: Both adaptive and non-adaptive online courses can be evaluated with the user measurements. In either case, high performance may indicate that the course was effective. However, the multiple paths through the material in the case of an adaptive course should also ensure a consistent user experience [4]; high-performing users do not necessarily indicate that the adaptation mechanism succeeded. A metric must thus incorporate consistency as well as an efficacy score.

C3. Online computations: Users generally take weeks or months to complete an online course, which can result in long evaluation cycles if the metric value can only be computed once the course has ended (*e.g.,* with A/B testing or surveys). A metric that can be computed efficiently and regularly updated as users progress through the course is desirable. This online capability would allow instructors to receive feedback as the course progresses, giving them a better opportunity to address weaknesses revealed before the course completes.

1.2 Our Contributions

In this work, we formulate a mathematical framework for metrics that address challenges C1-C3. Our framework takes as input a set of user characteristics derived from observed data of an online course, and we quantify several example characteristics (*e.g.,* path deviation, engagement). To demonstrate our solution, we leverage data from a course that we hosted for Velocity Chess, a popular online chess competition platform that teaches users techniques for playing the game. With this dataset, we compare a comprehensive statistical analysis of the course data with an instance of our metric, and show that the metric provides insights that are difficult to glean from the analysis alone.

More specifically, our work answers the following questions:

(i) How to define metrics that addresses the three challenges? We begin in the next section by presenting our metric framework. To address C2, it includes statistical factors for (i) the consistency of learning characteristics over different users and course units, and (ii) the overall efficacy of the course as indicated by the actual characteristic values. In doing so, to address C1, we account for the fact that different quantities may have different relationships with efficacy; for example, while efficacy is generally linear in quiz performance, *i.e.,* higher performance is a positive indicator, the relationship with time spent is concave, *i.e.,* excessively high time spent indicates confusion. Our metric parameters can also be flexibly chosen to consider different subsets of the quantities, and to induce different priorities on consistency and efficacy. Finally, given the fine-granular timescale at which certain types of learning data are captured, the metric can be computed at any point in the course, addressing C3.

(ii) How to quantify characteristics to be assessed by the metrics? After presenting the metric framework, we derive formulas for several learning quantities that characterize user actions associated with efficacy in a course. We consider three categories of quantities in particular: *behavioral* (*e.g.,* engagement and time spent on content), *performance* (*e.g.,* quiz scores and knowledge transfer), and *navigation* (*e.g.,* deviation from recommendations). While the exact formulas we present for these quantities are specific to the data capture formats of our system, they are readily extensible to other collection mechanisms and content formats too. In performing a statistical analysis of our dataset in terms of these quantities, we observe that (i) while behavior and performance tend to increase throughout the course, they exhibit high variance in different units, and (ii) little correlation exists between most quantities. (i) and (ii) indicate potential room for improvement in terms of efficacy and consistency, respectively.

(iii) How do the insights of the metrics compare to those revealed through full statistical analyses? We then evaluate an instance of our metrics on this dataset, and compare the findings to those of the more comprehensive statistical analysis. Our metric shows that (i) 50% of the users attain less than 16% of the maximum observed metric value, and (ii) a considerable number of users are highly engaged in the course, but performance tends to be low. Both insights are consistent with the findings from the statistical analysis. Additionally, we find that the metric output contains more insight into learner attrition rates than do other course quantities. Overall, we find that our metric can successfully quantify course consistency and effectiveness, giving instructors straightforward statistics that allow them to improve future versions of the course.

We finally review related work on metrics for online courses and recommendation platforms more generally, and then discuss implications and extensions of the work before concluding the paper. In particular, though our metric is designed for adaptive online courses, it is applicable to any personalized recommender system in which multiple signals can give insight into efficacy.

2. OUR COURSE METRIC FRAMEWORK

In this section, we present our metric framework for evaluating adaptive online courses. We first formalize the general architecture of adaptive courses and then specify the combination of consistency and efficacy mathematically.

2.1 Course Architecture and Metric Input

We assume that any adaptive course is organized into a set of units $\mathscr{U}$, with $u \in \mathscr{U}$ denoting a particular unit u. Within each u there can be one or more content files that a user is expected to study, *e.g.,* videos or PDF documents. At the end of u, there may be an assessment quiz consisting of a series of questions. We assume that the course captures user behavior while interacting with the content in u as well as user performance on the corresponding quiz.

Generally speaking, the adaptation logic of the course will recommend for each user a sequence of units $U_r = (u_r(1),...,u_r(t_r))$ to visit, with $u_r(i) \in \mathscr{U}$ denoting the one recommended at time i. This may be different than the actual chronology $U_a = (u_a(1),...,u_a(t_a))$ of the units that the user chooses to visit. The determination of $u_r(i)$ may in general be based on analysis of the user's actions in $u_a(1),...,u_a(i-1)$, including but not limited to their behaviors from interacting with the content, their performance on the quizzes, and potentially sources of data external to the course that are available to the system. Note that certain units may appear multiple times in U_r or U_a, as users may or may be recommended to repeat/revisit one or more units.

2.1.1 Quantities Q

Our metric takes as input a set of characteristics regarding users in the course to be jointly assessed, which we refer to as the set of quantities Q. Each quantity $q \in Q$ can belong to one of at least three categories: behavioral, performance, or navigation, with the latter involving differences between U_a and U_r. The instructor can choose (i) which characteristics are to be used as quantities in Q, and (ii) whether each q is for a particular unit u or across all units in the course. For instance, Q could be just time spent T_u in a single unit u, or $Q = \{T_1, T_2, ..., g_1, g_2, ...\}$ could be the time spent T_u and assessment grades g_u over all units u in the course.

In this way, the quantities are representative of the (heterogeneous) user feedback to be analyzed by the metric. We discuss the definition of particular quantities for our dataset and data capture system in the next section.

2.2 Distribution-based Metric Framework

The metric framework must use the quantities to determine course consistency and efficacy.

2.2.1 Quantifying Consistency

We incorporate a measure of consistency through the distribution of the quantities Q over users. We construct this distribution over a discretized set of possible quantity combinations, *i.e.,* all feasible combinations of quantities that users could exhibit.

Formally, let $\mathscr{X}$ denote the support of the distribution, *i.e.,* the set of feasible outcomes (note that our empirical samples may cover only a subset of the theoretically feasible outcomes). Further, let $x = (x_1, x_2, ..., x_{|Q|}) \in \mathscr{X}$ be a particular point in the support, with x_q being the value of quantity q at this point. The empirical cumulative distribution function (CDF) $F_Q(x)$ over the set of quantities Q is then obtained as $F_Q(x) = \frac{1}{|\mathscr{X}|}\sum_{y\in\mathscr{X}} \mathbb{1}\{y_q \leq x_q \;\; \forall q\}$ along with the associated probability distribution function $f_Q(x)$. Here, $\mathbb{1}$ is the indicator function, and since $f_Q(x)$ is defined over a finite support we have $\sum_{x\in\mathscr{X}} f_Q(x) = 1$.

We wish for the consistency measure to be maximized when the distribution $f_Q(x)$ is concentrated at a single point. To this end, we define the consistency measure

$$M_Q^c(\mathscr{X}) = \sum_{x\in\mathscr{X}} h\left(f_Q(x)\right)$$

where h is a differentiable, strictly convex function on $[0,1]$ with $h(0) = 0$ (no density at x should map to no change in the measure). Strict convexity of h ensures that as density is distributed across more points, the consistency $M_Q^c(\mathscr{X})$ will decrease, a property that we prove formally in our online technical report (see **Proposition 1**) [7]. We could set $h(x) = x^2$, for example.

2.2.2 Combining Efficacy and Consistency

The consistency measure $M_Q^c(\mathscr{X})$ does not carry any information about efficacy: it can be maximized if users concentrate at any point $x \in \mathscr{X}$, regardless of how effective the course is for users at that point. Our metric framework must also incorporate the actual quantity values x_q. To do this, we modify $M_Q^c(\mathscr{X})$ by scaling the $h(f_Q(x))$ by a function of the observed x_q:

$$M_Q^s(\mathscr{X}) = \sum_{x\in\mathscr{X}} \sum_{q\in Q} z_q(x_q) h\left(f_Q(x)\right) \tag{1}$$

We suppose that $z_q(x_q) \geq 0$ for each $x \in \mathscr{X}$. Different choices of the function z_q can then put greater or lesser emphasis on consistency over quantity monotonicity.

Choosing z_q. For a given distribution $f_Q(x)$, M_Q^s is monotonically increasing in $z_q(x_q)$ for each x_q. While different values of x for a given individual user would change the estimated distribution f_Q, we suppose that there are sufficiently many users that these changes are small and do not affect M_Q^s's overall monotonicity. The function z_q must therefore be chosen separately for each quantity q to map more effective x_q to a higher $z(x_q)$.

For quantities that are monotonically related to course effectiveness, *e.g.,* quiz performance, we can take $z_q(x) = x$. Most of the quantities q we consider in this work fall into this category, but two of them do not. The first is time spent: a course is ineffective for users who spend an excessively short or long amount of time on it [1, 2]. The second is deviation from the adaptive course's recommended path: some deviation from the recommended path can be helpful, particularly to review additional content, but an excessive amount indicates the adaptation is not meeting users' needs. Thus, if q represents either of these quantities, we should take z_q to be a function that initially increases with x_q and then decreases, *e.g.,* a gamma function.

The z_q must also have a component to adjust how much we wish to emphasize consistency compared to monotonicity. For instance, if we define $z_q(x_q) = (1 + x_q)^\alpha$ for the parameter $\alpha \in [0, \infty)$, then at $\alpha = 0$ we would only consider consistency ($z_q = 1$). As $\alpha \to \infty$, the z_q term in M_Q^s would dominate the $h(f_Q)$ term, and a larger concentration of users at a more effective point $x \in \mathscr{X}$ would result in a larger marginal increase in M_Q^s, when compared with the increase at a smaller value of α. Thus, for larger values of α, the metric would attain a greater value if a few users have a very positive experience, compared to if all users have a consistent, moderately positive experience. We formally quantify this insight in our online technical report (see **Proposition 2**) [7] by considering, for each value of α, the set of quantity values x for which a consistent experience concentrated at x yields a higher metric value than an inconsistent, uniform distribution of user characteristics over the entire set of feasible quantity values $\mathscr{X}$.

3. DERIVING QUANTITIES FROM DATA

In this section, we derive several specific quantities from learning data that can form the set Q in our metric framework. We do so based on data formats from our course delivery system, considering the case of an adaptive online course we hosted for Velocity Chess, an open chess competition website. We will categorize user activities into three main quantity types: navigation, behavioral, and performance.

While formulating the quantities, we also perform a comprehen-

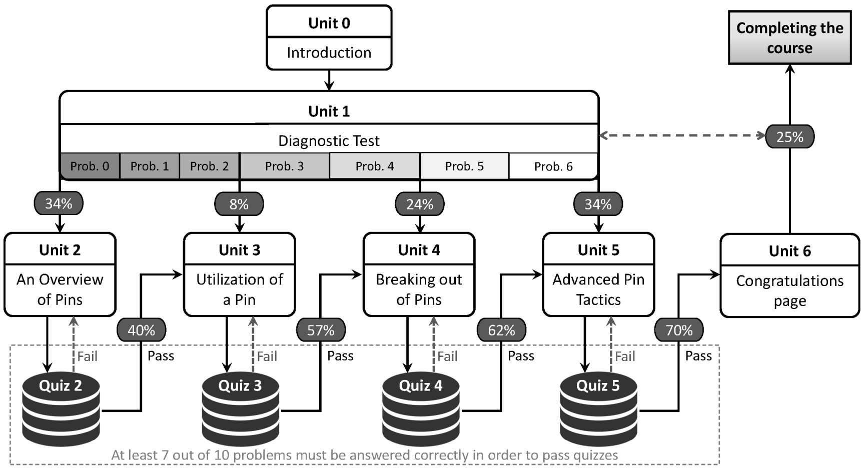


Figure 1: The course consists of seven units: a welcome unit (unit 0), the diagnostic test (unit 1), four core units (units 2-5), and a completion page (unit 6). The adaptation logic is also indicated in the diagram. The percentages indicate the fraction of times that recommendation was made, *e.g.,* in unit 3, a user will answer the quiz and be recommended to advance to unit 4 57% of the time (as opposed to failing the quiz or dropping off before finishing the quiz).

sive statistical analysis of the dataset. In doing so, we make three main findings: (i) many users deviate significantly from their recommended paths, (ii) there is high variability in user behavior and performance, and (iii) user activity and performance tend to increase later in the course. In the next section, we will see that our metric framework also reveals these insights.

Statistical tests. In certain cases, we will run statistical tests to compare distributions of quantities so as to derive qualitative insights into the course efficacy. For these, we will report the p-value (p) and the corresponding test – Wilcoxon Rank Sum (WRS), F-test of Variance, or Pearson correlation [21] – in the description.

3.1 Course Structure and Data Capture

The course we analyze teaches users the Pins strategy for playing chess, from beginner to advanced levels, individualizing the material based on the user's inferred level. It was open to all site users starting in December 2015; we consider the data collected over the one-year time period from December 2015 to 2016, comprising 4,877 enrolled users.

The course architecture and adaptation logic are defined in Figure 1. The content is divided into six units $u = 0,...,6$. The core material of the course is contained in Units 2-5, which are of increasing difficulty. Each of these "core units" is comprised of a series of slides and ends with a quiz; after completing the quiz, the course's adaptation logic directs users to a new unit based on their quiz performance. For instance, an average performer may be recommended to proceed to the next unit, but a user who failed the quiz may be asked to repeat that unit. Unit 1 is a diagnostic test that all users take, based on the results of which the adaptation will recommend a core unit to start at.

Clickstream event capture. Each slide in the course is either video-based or text-based. For video slides, the user has a scroll bar to navigate the video, and all `playback` events are captured by the system; these consist of `pause`, `play`, `scrub` (either forward or backward), and `replay` (*i.e.,* starting the slide over), together with the position of the video at which the event occurs. For text slides, there is a single playback event when the user accesses it. In both cases, a slide `change` event is generated when the user moves to a new slide. Slide IDs and UNIX timestamps of all events are also recorded; the IDs include both the previous (immediately before event) and next (immediately after) slides, which differ for `change` events.

The system also records user navigation events independent of particular units: unit `enter` and `exit`, course `login` and `logout`, and application `foreground (fgnd)` and `background (bgnd)`, *i.e.,* when the application is the current active tab on the user's computer. Using these events, we are able to infer a user's navigation between units and their behavior within units. For their quiz performance, we use the `response` events that the system collects after a user answers a question, indicating whether the answer was correct or not.

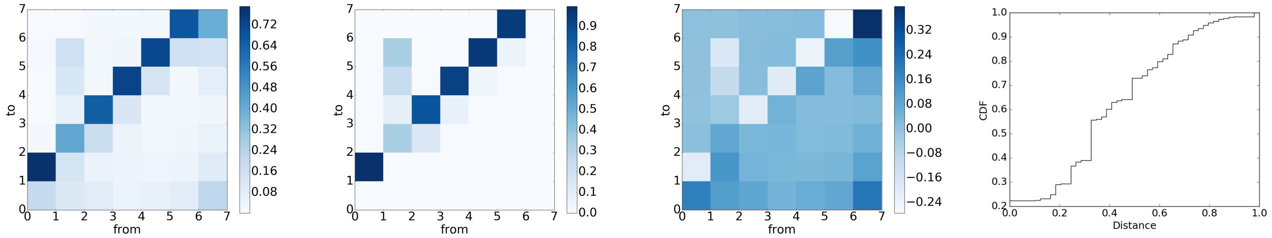
3.2 Quantifying User Navigation

We first investigate user progression through the course units, and use that to define a navigation quantity. Recall that while the system itself generates an adaptation path U_r for each user, the user's chosen path U_a may deviate from the system recommendation. We count a unit as "visited" in U_a if the user spent at least 5 seconds on the material in the unit; time spent on the unit's material is itself a quantity defined in later sections.

Unit-to-unit transitions. 2,186 out of 4,877 users entered the diagnostic test (unit 1) from the introduction (unit 0). For subsequent units, the percentages in Figure 1 summarize the users' recommended paths U_r:

Skill branching: Of the 1,310 users who completed the diagnostic test, the majority (68%) were placed either at the most beginner or the most advanced level. This heterogeneity is common in MOOCs.

Repeating vs. advancing: When placed in core unit u, the fraction recommended to advance to $u+1$ as the next step increased in u (40% to 70%). As users get further through the course, they are more motivated to finish (25% of those who accessed the diagnostic test ended up finishing). Interestingly, very few users are



(a) User navigation.

(b) System recommendation.

(c) Difference (a)−(b).

(d) Distance CDF.

Figure 2: Comparison between (a) user navigation and (b) recommended navigation between units. A point (j,i) in the diagram is the fraction of times unit j was selected while starting on unit i. (c) gives the difference in fractions, illustrating a strong deviation between actual and recommended transitions between units. This is supported by (d) the empirical CDF of the Levenshtein distance d between actual and recommended sequences.

recommended to repeat the core units (less than 3.7% in each case). The remaining users dropped out; we will investigate drop-off further in the next section.

Figures 2a-c show the discrepancies in unit-to-unit transitions between user behavior U_a (a) and system recommendations U_r (b), with the difference between the fractions plotted in (c). In the core units, the vast majority of recommendations are to advance from u to $u+1$, as discussed above. Users' actual paths, on the other hand, are more diverse: there are visibly more repetitions than the system recommends, and also occasional skips back to prior units. Thus, many users likely feel the need for more course content review than is recommended.

Path deviation quantity. We quantify navigation as users' deviation from their recommended paths through the course. To do this, recall the notation $U_r = (u_r(1), ..., u_r(t_r))$ and $U_a = (u_a(1), ..., u_a(t_a))$. For this course, we always have $t_a \geq t_r$ because navigation can only add steps to the recommended path; users cannot skip units unless recommended. From this, we define the path deviation quantity

$$d = \frac{1}{|U_a|} v(U_a, U_r)$$

where $v(\cdot)$ is the Levenshtein (edit) distance between the two sequences [26]. We choose Levenshtein rather than other distance metrics, *e.g.,* longest common subsequence, because it allows for insertion, deletion, and substitution operations in between strings. In our application, insertion captures users adding additional revising units into U_a from U_r, and substitution captures them choosing to visit different units than those recommended. Division by $|U_a|$ ensures that $d \in [0,1)$.

Figure 2d gives the cumulative distribution function (CDF) of the quantity d over users in the dataset.[1] The mean deviation is 0.36, which can be interpreted as user paths being 36% different from the recommendations on average. On the one extreme, about 22% of users follow the recommendations exactly (*i.e.,* $d = 0$), while on the other hand, 25% of users deviate by 56% or more.

3.3 Quantifying User Behavior

We derive three quantities of user behavior within units: time spent, completion rate, and engagement.

3.3.1 *Defining Behavioral Quantities*

Let $E = (e_1, ..., e_n)$ be the sequence of n clickstream events generated by a user in the course. For each event e_k, let $s(e_k)$ denote its next slide ID, *i.e.,* the ID immediately after. We write $s \in S_u$ to denote that slide s appears in unit u.

Time spent. Let $t(e_k)$ be the timestamp of event e_k. The time registered for the interval between e_k and e_{k+1} is:

$$T_k = \begin{cases} \min\left(t(e_{k+1}) - t(e_k),\ \tau\right), & \text{if } e_k \neq \texttt{bgnd} \\ 0, & \text{otherwise} \end{cases}$$

In other words, we do not consider time intervals for which the app is in the background, and set the parameter $\tau = 600$ seconds to upper bound the time between actions, capping excessively long intervals when the user likely walked away. From these intervals, the time spent on slide s, T_s, and the time in unit u, T_u, are

$$T_s = \sum_{k:\, s(e_k)=s} T_k, \qquad T_u = \sum_{s \in S_u} T_s,$$

since $s(e_k) = s$ implies that T_k is time spent on s.

Completion rate. Completion of slide s is a binary measure, defined as $R_s = 1$ if $T_s \geq \varepsilon$ and 0 otherwise. We set $\varepsilon = 5$ sec so that if the user spent at least 5 seconds on s it is considered completed. From this, the completion rate of unit u is defined as

$$R_u = \frac{1}{|S_u|} \sum_{s \in S_u} R_s,$$

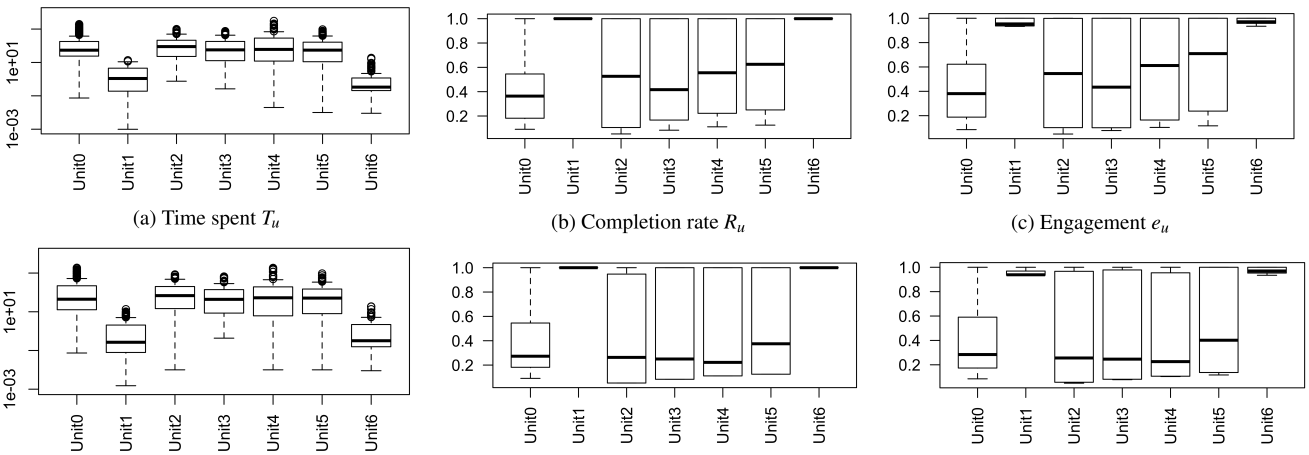
where $|S_u|$ is the number of slides in u. Note that R_u is between 0 (no slides completed) and 1 (all completed).

Engagement. Let $\bar{T}_s$ be the "expected" time spent on slide s. Following the method proposed in [6], we calculate the engagement of a user on unit u as

$$e_u = \min\left(\gamma \times R_u \times \prod_{s \in S_u} \left(\frac{1 + T_s/\bar{T}_s}{2}\right)^{\alpha},\ 1\right).$$

Here, $\alpha \geq 0$ models the diminishing marginal return on time spent, *i.e.,* more time spent on the same slide counts incrementally less towards engagement. The division by 2 makes the computation relative to a user who spends the expected $T_s = \bar{T}_s$ on each slide. $\gamma \in (0,1]$ is a constant that controls the overall spread of the distribution; a user who registers the expected time spent and 100%

[1] In this plot, we only consider users with $|U_a| > 2$, *i.e.,* those who proceeded past the diagnostic test.



(a) Time spent T_u

(b) Completion rate R_u

(c) Engagement e_u

(d) Time spent T_u (repeating)

(e) Completion rate R_u (repeating)

(f) Engagement e_u (repeating)

Figure 3: Distributions of time spent, completion rate, and engagement across units in our dataset. Each quantity is considered both (i) for all user visits to a unit in a-c and (ii) for all visits past the first one (*i.e.,* repeating) in d-f. The core units 2-5 each exhibit significant variation in user behavior.

completion on each slide will have $e_u = \gamma$. By default, we set $\gamma = 1$, $\alpha = 0.1$, and $\bar{T}_s = 60$ sec.[2]

All three behavioral quantities – time spent T_u, completion rate R_u, and engagement e_u – have been defined here on a per-unit basis. We also consider them at a course level to get a complete picture of overall behavior. For these details, see online technical report [7].

3.3.2 Behavioral Analysis

Figure 3 gives boxplots of the three behavioral quantities in our dataset, across units. For each quantity, we show behavior over all user visits to units, as well as repeating visits only.

We first observe that *behavior in the core units exhibits high variation in each of the quantities.* The interquartile ranges (IQR) of R_u and e_u are between 0.75 and 0.90, out of a maximum range of 1.0. The ratio of the IQR to the median – a non-parametric coefficient of variation [21] – is larger than 1.2 in each case, up to 4.6 for time spent in unit 4. The IQRs for time spent are up to 275 sec.

Also, *user activity tends to increase in later core units* (WRS $p \leq 0.033$). While time spent (T_u) is reasonably consistent in units 2 to 5 – with medians around 60 sec – completion rate (R_u) and engagement (e_u) both increase considerably from units 3 to 5. In particular, the median R_u rises from 0.42 to 0.63 and the median e_u increases from 0.43 to 0.71. The WRS p-values associated with these changes are significant ($p \leq 0.033$) in each case. Combined with the consistent values of T_u, this implies that users are distributing their time more evenly across slides in later units. This is somewhat surprising because the later material is more challenging, so we would expect certain slides to require more time.

For repetitions, the median t_u drops by < 25 seconds, while R_u and e_u drop more substantially, from 0.17 to 0.39 depending on the unit. The small drops in time spent indicate that users spend a significant amount of time repeating. Coupled with large declines in completion rate, this implies that overall, users are focusing on a more specific set of slides while repeating. Large variations in behavior, however, remain: the third quartiles of R_u and e_u barely move at all.

3.4 Quantifying User Performance

We derive two quantities for user performance: quiz performance and earned virtual currency (called vChips).

3.4.1 Defining Performance Quantities

Quiz performance. Let $\mathcal{N}_u = \{n_1, n_2, \ldots\}$ denote the set of questions in the question bank for unit u. Upon a user's lth visit to the quiz for u, they will be given a random subset $\mathcal{N}_u^l \subset \mathcal{N}_u$ of these questions to answer. The number of points earned on the lth visit to u is calculated as $p_u^l = \sum_q p_q^l$, where $p_q^l = 1$ if the user answered question q correctly on the lth attempt, and 0 otherwise. The total points earned on u is then $p_u = \sum_l p_u^l$, and the total points earned in the course is $p_c = \sum_u p_u$. From this, the user's quiz grade on u, g_u, and grade in the course, g_c, are

$$g_u = p_u / N_u, \qquad g_c = p_c / N_c,$$

where $N_u = \sum_l |\mathcal{N}_u^l|$ is the total number of questions answered by user in unit u, and $N_c = \sum_u N_u$ is the total number given to the user in the course. In this way, g_u and g_c are between 0 (no points received) and 1 (all questions answered correctly). Note that, due to question randomization and course adaptivity, $\mathcal{N}_u^l$, N_u, and N_c will vary for each user.

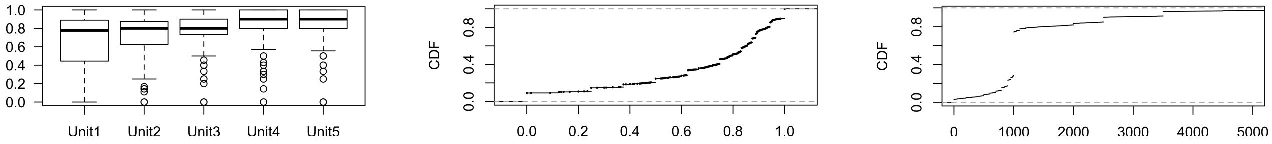
vChips. Velocity Chess awards users vChips[3] – a form of virtual currency – based on their activity and performance on the site. The vChips can be obtained by winning chess games, winning prizes in tournaments, finishing daily challenges, and correctly solving chess puzzles. They can thus measure players' chess skill in practice.

3.4.2 Performance Analysis

Figure 4 gives the distributions of the performance quantities g_u, g_c, and vChips. Boxplots of g_u are shown in (a) for each unit that has a quiz, while CDFs of g_c and vChips are given in (b) and (c).

[2] 1 minute is the approximate median of time spent on each slide in the dataset.

[3] https://www.velocitychess.com/faq



(a) Unit quiz performance g_u

(b) Course quiz performance g_c

(c) vChips

Figure 4: Distributions of quiz grades across units, quiz grades across the course, and vChips for users in our dataset. Quiz performance improves in later units, exhibiting significant variation throughout, though less-so than the behaviors in Figure 3. The vChips have a high concentration around 1,000 chips.

Just as user activity increased in later units, we find that *user quiz scores increase further into the course* (WRS $p \leq 0.026$). The median grade in (a) rises monotonically from 0.78 in unit 1 to 0.9 in unit 5. Despite the increase in difficulty, the users reaching later units are likely more knowledgeable and can thus perform better.

We also find that *users' performance is less variable than their behavior* (F-test $p \leq 5.19 \times 10^{-3}$ with the exception of T_u): the IQRs for unit grades g_u range from 0.20 to 0.44, with corresponding IQR-to-median ratios between 0.22 and 0.57. These ratios are smaller than those observed in Figure 3. The vChips have even less variation: with a median of 1,000 and an IQR of 75 chips, the ratio is only 0.075. The vChips have a heavy tail as well, with the mean being 3,271.

3.5 Quantity Correlations

The above analysis indicates that there is high variability in users' behavior and performance quantities unit-by-unit as well as in their vChips and path deviation quantities over the full course. Taken alone, however, any one of these quantities fails to capture the diversity of users taking open online courses. Since our metric framework in Section 2 seeks to aggregate them into an overall measure of efficacy, we also considered the correlation between the different quantities, both between quantities of the same type (Sec. 3.5.1) and between those of different types (Sec. 3.5.2). Overall, we found that most of the quantities exhibit little correlation, i.e., each provides unique information on the diversity of users taking open online courses [11]. In this section, we will present the most interesting of these findings; for the full set of scatterplots and corresponding statistical analysis, see our technical report [7].

Normalizing behavioral quantities. To perform this correlation analysis, we consider each user's quantity values at the course level. To translate the three per-unit behavioral quantities – time spent T_u, completion rate R_u, and engagement e_u – to per-course, we sum all of these quantities over all units of the course for each user,[4] and then normalize over the number of units visited. For completeness, we also considered the number of units suggested by the adaptation algorithm. Formally, let $U_a' \subseteq U_a$ be the set of unique units visited by a user, and $U_r' \subseteq U_r$ be the set of units recommended. The normalized quantities are defined as

$$x_c^a = \frac{1}{|U_a'|}\sum_u x_u, \qquad x_c^r = \frac{1}{|U_r'|}\sum_u x_u,$$

where x_u denotes the quantity (T_u, R_u, or e_u) for unit u, as defined in Section 3. The normalization for x_c^a ensures that the R_c and e_c quantities still lie in $[0,1]$. x_c^r, on the other hand, will become larger than x_c^a when a user takes the initiative of visiting units that were not recommended, i.e., that they could have skipped.

3.5.1 Correlations within Quantity Types

Figure 5 plots the course-level behavioral quantities against one another, normalizing by actual path (x_c^a). We see immediately in Figure 5(a) that there is not a strong relationship between time spent T_c and completion rate R_c, with a Pearson correlation coefficient $r < 0.4$. Those with completion of 100%, in fact, have the highest variation in time spent, perhaps due to them viewing more slides: users' variation in the time spent on each slide would then accumulate over more slides, leading to higher overall variability.

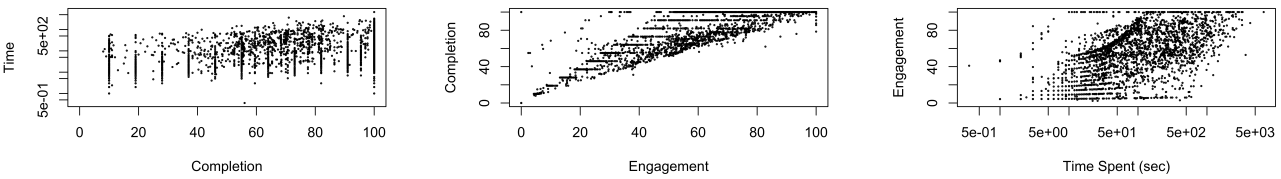
Figure 5(b), on the other hand, shows a strong positive correlation between completion rate and engagement e_c, with $r > 0.95$. This is expected since engagement is defined to be linear in R_u. Specifically, several users have moderate e_c and high R_c: they would have low T_c to pull the engagement level down. Figure 5 shows a positive correlation between e_c and T_c as well, though not as strong, and we can see cases where a low time spent corresponds to a moderate engagement value. Overall, we conclude that *though engagement is a combination of completion rate and time spent, each of the three quantities gives important information on user behavior.*

As for the performance quantities, Figure 6 gives a scatterplot of quiz score g_c against vChips. We see that *vChips and quiz scores are only weakly positively correlated.* The positive association is intuitive, because we would expect those answering the questions correctly to be more skilled in chess and thus to have the potential to win more games. On the other hand, the lack of strength is surprising. There are many uncontrolled factors outside of the course that could affect this, though, such as whether the strategy taught in the course (pins) is useful in a given situation.

3.5.2 Correlations Between Quantity Types

From analysis between quantity types, our key finding is that *the only significant correlation is a positive one between engagement and quiz score*, while the rest of the pairs – distance vs. engagement, vChips vs. time spent, and so on – only have minor associations, if any. This can be seen in Figure 7, which gives scatterplots of selected pairs – vChip and engagement in (a), quiz and engagement in (b), and quiz and distance d in (c) – with behaviors normalized by recommended path (x_c^r). The scatterplot in (b) has a correlation coefficient of $r > 0.75$, meaning that users who complete more slides and/or spend more time on each slide tend to have improved quiz scores. Figure 7(a), on the other hand, shows that *users' vChips are only weakly positively correlated with their behavior*: users with higher engagement do tend to have slightly more vChips, but there are still many instances of low engagement users earning among the most vChips (potentially those with prior knowledge of the pins tactic) and users with high time spent earning the least vChips (potentially those who struggle with the course).

[4] Given the variability between units observed in Section 3, we consider per-unit, per-user quantities in Section 2.



(a) Time spent and completion ($r = 0.395$)

(b) Completion and engagement ($r = 0.959$)

(c) Engagement and time spent ($r = 0.490$)

Figure 5: Scatterplots of the behavioral quantities, normalized by the number of units visited (*i.e.,* x_c^a). The correlation between completion and engagement is strong, but weaker for the other two pairs.

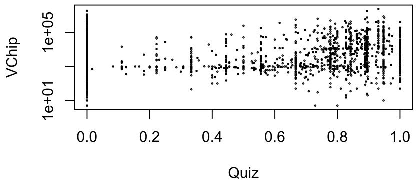


Figure 6: Scatterplot between the performance quantities, vChips and quiz score g_c. There is not a strong correlation between them ($r = 0.138$).

We also found *a weak negative correlation between distance and the behavioral and performance quantities*; the case of distance and quiz score is plotted in Figure 7(c). Users who followed the adaptation algorithm's recommendations, then, have a mild tendency to be more engaged, spend more time, and obtain higher grades than those who deviate from the recommendations. On the other hand, a greater deviation can still lead to lower course activity and grades for some users, and there are different users over the full range of possible completion rates, engagement, and time spent that cover the full range of possible distances. This emphasizes again that the navigation quantity conveys different information than the performance and behavioral quantities.

4. METRIC EVALUATION

The statistical analysis in the previous section revealed that while activity and performance tend to increase further in the course, there is high variability in the quantities overall, and thus room for improvement in consistency and efficacy. In this section, we first perform an evaluation of the course using our proposed metric framework, and show that it also leads to these conclusions. We then consider course drop-off rates, and find that our metric yields better insight into this than do the quantities.

4.1 Course Consistency and Efficacy

Before presenting the results, we first specify particular inputs and parameters of M_Q^s in (1), as well as a sampling procedure to aid in the quantity distribution estimation.

Input quantities Q. The input to M_Q^s is user data on a set of quantities Q. Based on the definitions in the previous section, the full set of quantities Q takes each quantity at the unit-level except distance d and vChips which are only defined over the entire course, *i.e.,* $Q = \{\{e_u, R_u, T_u, g_u\ \forall u\}, d, \text{vChip}\}$. We also consider different subsets of Q in our evaluation, e.g., behavior quantities only.

Functions z_q and h. M_Q^s requires $z_q(x)$ and $h(x)$ for efficacy and consistency. For all metric variations, we take $h(x) = x^2$. We use $z_q(x) = x$ when q is an engagement e_u, completion rate R_u, performance g_u, or vChip quantity, as higher values of these quantities generally indicate a more effective course. We use the gamma distribution $z_q(x) = \frac{1}{\Gamma(k)\theta^k} x^k e^{-\frac{x}{\theta}}$ for the distance d and time spent T_u quantities, reflecting the non-monotonic relationship of these quantities with the course efficacy. We choose θ and k as the squared root of the median value of each quantity, so that g_q attains its maximum value at the median.

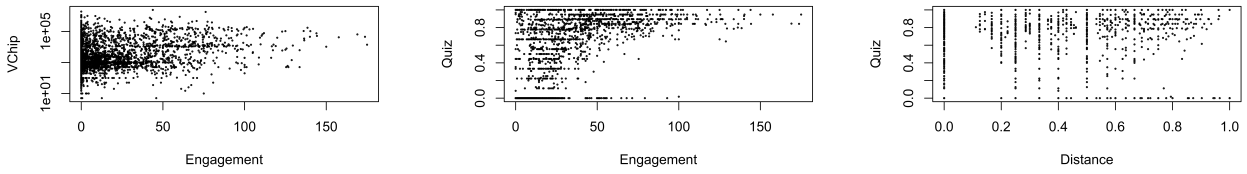
Sampling for $f_Q(x)$. To estimate the distribution $f_Q(x)$ of possible metric values, we first perform random sampling on the realized values of Q to better estimate the properties of the metric output. Similar to bootstrapping [8], for $q \in Q$ we uniformly at random sample non-zero quantity values $x \in x_q$ for each of the users. We take only nonzero values since zero quantity values correspond to inactive users, who may have dropped out of the course or skipped that unit. We take 100 different samples, and combine each with the original dataset to estimate the distribution f_Q and in turn calculate the metric values M_Q^s.

4.1.1 Results and Discussion

Our evaluation results of M_Q^s for the full quantity set as well as subsets are given in Figure 8. Each circle in each distribution plot of Figure 8 represents the metric value from one sample. These plots are the subject of the following discussion.

All quantities. We first consider the metric values over all units and quantities Q. Figure 8a shows the distributions for M_Q^s across samples. We see that many (roughly 50%) of the samplings yield fairly low metric values that are < 1. Considering that roughly 20% of the samples have an output of 6 or higher, meaning that a majority of cases yield less than 17% of the maximum value, this indicates *room for improvement in terms of overall efficacy and consistency*, as we concluded from the statistical analysis. Other samples show clear concentrations around 4 and 6, perhaps due to different quantities concentrating at these values. We also further analyzed the metric in terms of its two constituent pieces – actual quantity values and user consistency – to see whether one had a larger bearing on these low metric values. In doing so, we found that both contribute to low values, confirming room for improvement in both areas; for the corresponding plots, see our online technical report [7].

Behavioral vs. performance quantities. We next compare the metric outputs for the behavior and performance quantities only, in Figures 8b and 8c respectively. Since these quantities reflect different aspects of user activities, we would expect their metric distributions to differ, and we see that this is indeed the case. Also, we

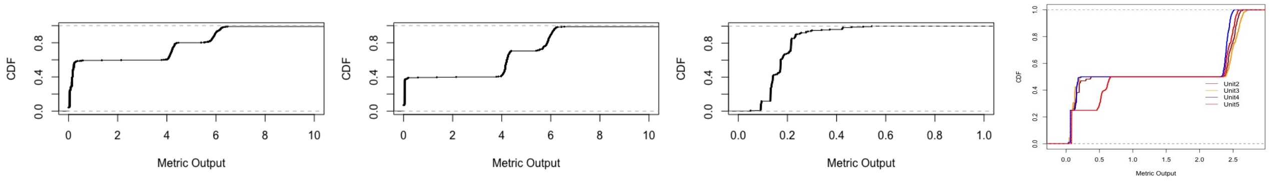


(a) vChip and engagement ($r = 0.129$)

(b) Quiz and engagement ($r = 0.758$)

(c) Distance and quiz ($r = -0.131$)

Figure 7: Scatterplots between selected quantities, with the Pearson correlation coefficient (r) reported for each. Behaviors are normalized by the recommended path (x_c^r). Most of the pairs of quantities exhibit little correlation.



(a) All quantities.

(b) Behavioral quantities.

(c) Performance quantities.

(d) Unit-by-unit quantities.

Figure 8: Distributions of the metric values for 100 different samples. Each circle represents one sample of possible values of Q. (a) is the CDF of M_Q^s considering all quantities. (b) and (c) are the distributions of the metric considering behavioral and performance quantities separately. (d) are distributions of metric values M_Q^s (1) for each unit, in which Q is taken to be each individual quantity. The distributions have a consistent shape for each unit, with over 80% of users experiencing low metric values.

observe that *the metric values are more varied for behavior than they are for performance*, which is consistent with our finding of high variability in behavioral quantities from the statistical analysis. Most users' performance metric values are low, concentrating around 0.2, suggesting poor performance and/or little user consistency. Recalling from Figure 4 that many users performed well on quizzes, this suggests that *these low metric values are likely due to low consistency in scoring*, rather than poor quiz scores. The behavioral metric values, on the other hand, suggest high behavioral quantities and/or high consistency in behavior. The high variability we observed in Figure 3 suggests that *effective behaviors contribute to these higher values*. This conclusion is consistent with the fact that several units show 25% of users obtaining the highest possible engagement and completion rates, whereas time spent is concentrated around its center.

Unit by unit quantities. To analyze differences between units, we also compute the metric over each individual quantity for each core unit. The results are shown in Figure 8d. We see that the distributions are fairly similar for units 2 to 4, exhibiting a fairly wide range of values in each case. As in the distributions over the full course in Figure 8a-c, there is a large concentration of metric values around smaller values, particularly 0. However, the maximum metric values are around 2.5, indicating that some users do have an effective experience in certain units. Indeed, users in unit 5 tend to have the highest values, with roughly 75% of them > 0.5. This is consistent with the conclusion from the statistical analysis that *user activity and performance tend to increase further in the course*.

Overall, these findings indicate that the course is effective at engaging users (Figure 8b), but – at least based on quizzes and vChips – there is room for improvement in teaching them how to play chess (Figure 8c). Given the free and open nature of Velocity Chess's platform, many users likely took the course more out of interest in chess and less out of a desire to memorize chess strategies, which may explain why users' performance is more inconsistent and less indicative of an effective course than their behavior.

4.2 Course Drop-off

We finally validate our metric by comparing it to user drop-off statistics. High drop-off rates are a notorious issue facing open online courses today [3]; we saw in the statistical analysis that our dataset does face this problem particularly in the first three units.

In Table 1, we compare three sets of values across the different units: (i) mean values of behavioral quantities, (ii) metric calculations on the corresponding quantities, and (iii) drop-off percentages, defined as the percentage of users for whom this unit was the furthest visited. Recall that Figure 8d also illustrated the metric values for different units, showing each unit tending to exhibit low values, at least on average.

Overall, we find that *the metrics contain better insight into drop-off than do the behavioral quantities*. Unit 1 experienced a high drop-off while the behavioral quantities e_u and R_u in Units 0 and 1 were fairly high. In particular, on average learners completed almost half of the content in Unit 0 and Unit 1, while almost half of the learners never proceeded past Unit 1. Such drop-off tendencies are difficult to observe from looking at the mean behavioral quantities in Table 1. The metric functions M_Q^s, on the other hand, tell another story; in particular, $M_{R_u}^s$ and $M_{e_u}^s$ in Units 0 and 1 are low when compared to the average values of R_u and e_u. We therefore conclude that when learners are highly likely to drop off, $M_{T_u}^s$ and $M_{R_u}^s$ tend to signal lower quality than do T_u and R_u.

On the other hand, we see that M_Q^s *and the behavioral quantities demonstrate similar trends in the second half of the course, where dropoffs are lower.* Looking at learner behavioral quantities after

	Item	Unit 0	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6
Mean Value	Time spent (T_u)	0.32	0.10	0.05	0.04	0.04	0.08	0.05
	Engagement (e_u)	61.6	46.9	9.82	7.40	8.97	10.44	15.71
	Completion rate (R_u)	41.89	42.16	8.36	6.59	7.63	9.79	14.15
Metric Value	Time spent ($M^s_{T_u}(\mathscr{X})$)	0.04	0.03	0.05	0.05	0.05	0.05	0.03
	Engagement ($M^s_{e_u}(\mathscr{X})$)	21.10	22.31	4.14	4.33	3.86	3.78	5.90
	Completion rate ($M^s_{R_u}(\mathscr{X})$)	4.14	15.12	4.33	4.39	4.07	4.24	5.40
	Drop-off	0.4%	45.0%	14.9%	7.1%	4.0%	5.2%	–

Table 1: Metric comparison with quantities and drop-off rates. The first row of the table entries gives the average learner behavioral quantities, the second row gives the metrics M^s_Q on the corresponding behavioral quantities, and the third gives the drop-off percentages. The low metric values in Units 0 and 1, compared with the corresponding behavioral quantity values, are consistent with these units experiencing high drop-offs.

Unit 2, we observe that T_u, R_u, e_u are generally low, as many learners fail to engage with the course content. The same trend can be observed in the metric values. Interestingly, however, the M^s_Q do tend to increase as the drop-off lessens from Units 3-6, even though our metric was not designed to incorporate this explicitly.

5. RELATED WORK

Learning and content analytics. Recent research in online learning has focused on developing analytics for instructors [2]. Machine learning techniques such as collaborative filtering and probabilistic graphical models have been applied to predict students' abilities to answer questions correctly [17, 23] or their final grades [16, 19]. Other studies have shown that student behaviors display patterns that are significantly associated with learning outcomes [2, 10]. User-content interactions and Social Learning Networks (SLN) have also been used to predict student dropoffs [18, 22], while SPARFA-Trace [13] was developed to track student concept knowledge throughout a course. Few works, however, have studied the efficacy of the course itself, our goal in this work.

Adaptive learning evaluation. Developing course efficacy metrics is particularly important for the growing number of adaptive online courses. For example, MIIC [4] and LS-Plan [14] are all adaptive course delivery platforms that support user- or system-defined individualization across different materials. We can use our metric to improve adaptation algorithms and user experiences. The two most common evaluation mechanisms for adaptive online courses are (i) A/B testing of adaptation versus control group and (ii) user surveys. Although A/B testing [4] allows researchers to test the effect of controlled variations, it is difficult to incorporate additional variables afterwards. Surveys can be used to supplement A/B testing [25], but these rely on user recollections and also cannot be computed at arbitrary points during the course. Our metric framework, in contrast, is easily applicable to different input variables and can be computed at any time during the course.

Online personalization metrics. Substantial amounts of research have been poured into online personalization for applications outside of education, particularly on recommendation systems that predict individual user preferences (see [5] for a survey). Traditionally, these systems have been evaluated with metrics like accuracy and RMSE on a holdout set. Yet these techniques have been criticized as being too distant from the actual user experience [15]. Therefore, newer metrics aim to incorporate factors such as diversity, novelty, and coverage [9,20]. Still, each of these metrics tends to focus on the final results of the prediction without taking into consideration users' prior and subsequent experience with the system. They are also difficult to apply to online courses, which aim to change users' internal knowledge states in ways that are not directly observable.

6. DISCUSSION AND CONCLUSION

We developed a metric framework for adaptive online courses that quantifies both the consistency of users' experiences in the course and the effectiveness of the course across multiple users. To measure effectiveness, we incorporated multiple quantities that describe the full range of user experiences, from their navigation through the adaptive course to their performance on quizzes and external tasks to their interaction with the course material. A statistical analysis of these quantities showed little consistency between different users' experiences and suggested that the course adaptation may not have been effective for many users: many users exhibited poor performance despite spending large amounts of time on the course, and others exhibited high performance but barely engaged with the material. Applying specific instances of our metric to the dataset showed that the metric contained many of the same insights as a statistical analysis, and revealed additional findings consistent with drop-off rates.

A full statistical analysis likely contains more insights than any single metric can provide. Defining a unified metric framework, however, not only allows us to more compactly represent a course's effectiveness, it also allows for direct, quantitative comparisons between different units of a course or even different iterations of a course. This information can then be used by an instructor to improve the material, either in the current or future offerings. While traditional A/B testing requires the instructor to vary one characteristic of the course at a time – which can be inefficient and result in an uneven course experience for different users – our approach enables instructors to estimate the marginal benefits of different interventions, allowing for more rapid and dynamic changes.

Our metric framework is not restricted to adaptive online courses: it can accommodate different quantities that may have distinct relationships to course effectiveness. Indeed, it can even be used for other types of personalized recommendation systems in which multiple quantities can give different insights into the recommendation effectiveness. For instance, users' ratings of a movie on Netflix may contrast with the time spent watching the movie, yielding contradictory information for the recommendation algorithm. Adaptive online courses are, however, perhaps more likely to exhibit such contradictory information than other recommendation settings, and online education presents other unique challenges that require the development of new metrics. The challenges of personalization in different applications motivate the consideration of such metrics more generally.

7. REFERENCES

[1] L. Breslow, D. E. Pritchard, J. DeBoer, G. S. Stump, A. D. Ho, and D. T. Seaton. Studying Learning in the Worldwide Classroom. *Research & Practice in Assessment*, 8, 2013.

[2] C. G. Brinton, S. Buccapatnam, M. Chiang, and H. V. Poor. Mining MOOC Clickstreams: Video-Watching Behavior vs. In-Video Quiz Performance. *IEEE Trans. Signal Proc.*, 64(14):3677–3692, 2016.

[3] 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 Trans. Learning Technol.*, 7:346–359, 2014.

[4] C. G. Brinton, R. Rill, S. Ha, M. Chiang, R. Smith, and W. Ju. Individualization for Education at Scale: MIIC Design and Preliminary Evaluation. *IEEE Trans. Learning Technol.*, 8(1), 2015.

[5] A. Calero Valdez, M. Ziefle, and K. Verbert. HCI for Recommender Systems: The Past, the Present and the Future. In *Proceedings of the 10th ACM RecSys*, pages 123–126. ACM, 2016.

[6] W. Chen, C. G. Brinton, M. Chiang, and D. Cao. Behavior in Social Learning Networks: Early Detection for Online Short-Courses. In *IEEE INFOCOM*, 2017.

[7] W. Chen, C. Joe-Wong, C. G. Brinton, L. Zheng, and D. Cao. Technical report. `https://tinyurl.com/ycm35c3a`, 2016.

[8] B. Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics*, pages 569–593. Springer, 1992.

[9] A. Gunawardana and G. Shani. A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. *Journal of Machine Learning Research*, 10(Dec):2935–2962, 2009.

[10] J. Kim, P. J. Guo, D. T. Seaton, P. Mitros, K. Z. Gajos, and R. C. Miller. Understanding in-video Dropouts and Interaction Peaks in Online Lecture Videos. In *Learning @ Scale*, pages 31–40. ACM, 2014.

[11] R. F. Kizilcec and E. Schneider. Motivation as a Lens to Understand Online Learners: Toward Data-Driven Design with the OLEI Scale. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 22(2):6, 2015.

[12] G. D. Kuh, N. Jankowski, S. O. Ikenberry, and J. Kinzie. Knowing what Students Know and can do: The Current State of Student Learning Outcomes Assessment in US Colleges and Universities. *Urbana, IL: University of Illinois and Indiana University, National Institute for Learning Outcomes Assessment (NILOA)*, 2014.

[13] A. S. Lan, C. Studer, and R. G. Baraniuk. Time-varying Learning and Content Analytics via Sparse Factor Analysis. In *SIGKDD*, pages 452–461. ACM, 2014.

[14] C. Limongelli, F. Sciarrone, M. Temperini, and G. Vaste. Adaptive Learning with the LS-Plan System: A Field Evaluation. *IEEE Trans. Learning Technol*, 2(3):203–215, 2009.

[15] S. M. McNee, J. Riedl, and J. A. Konstan. Being Accurate is not Enough: How Accuracy Metrics have Hurt Recommender Systems. In *CHI EA*, pages 1097–1101. ACM, 2006.

[16] Y. Meier, J. Xu, O. Atan, and M. van der Schaar. Predicting Grades. *IEEE Transactions on Signal Processing*, 64(4):959–972, 2016.

[17] 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. *JMLR*, 2011.

[18] J. Qiu, J. Tang, T. X. Liu, J. Gong, C. Zhang, Q. Zhang, and Y. Xue. Modeling and Predicting Learning Behavior in MOOCs. In *ACM WSDM*, pages 93–102, 2016.

[19] C. Romero, M.-I. López, J.-M. Luna, and S. Ventura. Predicting Students' Final Performance from Participation in Online Discussion Forums. *Computers & Education*, 68:458–472, 2013.

[20] A. Said and A. Bellogín. Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks. In *Proceedings of the 8th ACM RecSys*, pages 129–136. ACM, 2014.

[21] D. J. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. crc Press.

[22] T. Sinha, P. Jermann, N. Li, and P. Dillenbourg. Your Click Decides Your Fate. In *ACL EMNLP*, pages 3–14, 2014.

[23] A. Toscher and M. Jahrer. Collaborative Filtering Applied to Educational Data Mining. *KDD Cup*, 2010.

[24] R. M. Wachter. How measurement fails doctors and teachers. The New York Times, 2016.

[25] T.-C. Yang, G.-J. Hwang, and S. J.-H. Yang. Development of an Adaptive Learning System with Multiple Perspectives based on Students' Learning Styles and Cognitive Styles. *Educational Technology & Society*, 16(4):185–200, 2013.

[26] L. Yujian and L. Bo. A Normalized Levenshtein Distance Metric. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(6):1091–1095, 2007.

Student Performance Prediction by Discovering Inter-Activity Relations

Shaghayegh Sahebi
Department of Computer Science
University at Albany – SUNY
Albany, NY
ssahebi@albany.edu

Peter Brusilovsky
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA
peterb@pitt.edu

ABSTRACT

Performance prediction has emerged as one of the most popular approaches to leverage large volume of online learning data. In the majority of current works, performance prediction is based on students' past activities in graded learning resources (such as problems and quizzes), while their activities in non-graded resources (such as reading material) are ignored. In this paper, we introduce an approach that can take advantage of students' work with non-graded learning resources, as *auxiliary* data, in order to predict students' performance in graded resources. This approach can discover the hidden inter-relationships between learning resources of different types, only using student activity data. Based on our experiments, the proposed approach can significantly reduce the error of student performance prediction, compared to baseline algorithms, while discovering meaningful and surprising relationships among learning resources.

Keywords

student modeling, learning material correlation discovery

1. INTRODUCTION AND RELATED WORK

The learning data abundance, due to popularity of Massive Open Online Courses (MOOCs), introduces new opportunities and challenges for the educational data mining (EDM) field. On one hand, larger volumes of student data can increase performance of traditional EDM approaches. For example, a performance prediction approach that is popular in the area of intelligent tutoring systems, offers a good basis for learning personalization. If the data-driven performance model predicts that some problem will be solved by the current student with a high probability, this problem could be skipped in favor of a more challenging one. If the expected performance is low, students could be offered some help and supplementary material. MOOC-scale data can help improving performance prediction making this approach more usable. On the other hand, data coming from modern MOOCs is usually more heterogeneous and too complicated for traditional EDM approaches. Unlike conventional Intelligent Tutoring Systems (ITS), that are mostly based on problem-solving, MOOCs offer students to learn and assess their knowledge using a variety of learning resources, such as reading materials, lecture videos, assignments, exams, graded quizzes, and discussions. This leads to various types of learning activities for students. With that heterogeneity, come interesting challenges: how to use information about student work with diverse learning resources to assess student knowledge or predict student performance? what is the relationship between concepts that are offered in different learning resource types?

A number of research projects, focused on alternative learning resources, demonstrated that many kinds of resources could considerably contribute to student learning. For example, Najar et al. studied effect of adaptive worked examples versus unsupported problem solving and showed that adaptive worked examples can lead to faster and more effective learning [Najar et al. 2014]. Also, Agrawal et al. showed that enriching textbooks with additional forms of content, such as images and videos, increases the helpfulness of learning material [Agrawal et al. 2014]. This indicates that ignoring the interaction between various types of resources limits our understanding of students' learning behavior and the efficiency of mining and analytical tasks, such as student knowledge modeling or performance prediction. Additionally, understanding inter-relationships between different resource types and student activities can help instructors in having more well-informed decisions on their course design. Modeling such inter-relationships in students' data can provide a unified view to data heterogeneity and present a better understanding of student learning, by modeling these different resource types that present student activities.

While there are some studies in the literature on impact of various learning resources on learning, the relationship between learning resource types and their effect on predicting student performance is under-investigated. For example, Wen and Rosé studied student patterns across different activity types and concluded that these patterns can provide insights into different activity distributions between high-grade and low-grade students [Wen and Rosé 2014]. However, their goal was not to predict student grades from their activities. Velasquez et al. [Velasquez et al. 2014] identified learning aid use patterns using cluster analysis. They showed that high use of learning aids is significantly correlated with students' exam performance. But, they did not predict student performance. Sao Pedro et al. [Sao Pedro et al. 2013] extended Bayesian Knowledge Tracing by conditioning the learning on whether the students received scaffolding in a topic or not. This model uses extra context information (topics) in addition to student performance, does not discover the relationship between learning resource types, and does not distinguish between different learning resources. Jiří and Pelánek studied learning resource similarities [Jiří and Pelánek 2017], but it was on graded resources, not considering resource types, and not predicting student performance.

One reason for unpopularity of using heterogeneous resources for predicting student performance is their potential conflicting effects. For example, Beck et al. investigated if providing assistance (help) to students benefits them using experimental trials, Bayesian Evaluation and Assessment framework, and learning decomposition [Beck et al. 2008]. In their studies, experimental trials and learning decomposition showed that assistance hurts students' learning. However, the Bayesian Evaluation and Assessment framework found that assistance promoted students' long-term learning. More recently, Huang et al. discovered that adaptation of their framework (FAST) for student modeling by including various activity types may lead researchers to contradictory conclusions [Huang et al. 2015]. More specifically, they studied the impact of example usage on student learning. In one of their formulations student example activity suggests a positive association with model parameters, such as probability of learning, while in another formulation this type of activity has a negative association with model parameters. Also, Hosseini et al. concluded that annotated examples show a negative relationship with students' learning, because of a selection effect: while annotated students may help students to learn, weaker students may study more annotated examples [Hosseini et al. 2016].

Another complication for considering heterogeneous resources is the difficulty in interpreting students' observed activities. In graded resource types, such as assignments and quizzes, a student's score explicitly represents her knowledge on the topic. Whereas in other resource types, such as reading material, there is no direct evaluation or explicit observation of student's knowledge. Hence, measuring the effect of such learning resources on students' knowledge, and thus predicting their future performance, would be a challenging task.

In this paper we propose an approach motivated by canonical correlation analysis (CCA) to discover the interaction between different learning resource types, using student activities, and to predict student performance on different learning resources. Our proposed approach can uncover latent relationships among subsets of learning recourses from different types and can quantify these relationships. Our analysis on two real-world datasets demonstrates that the discovered relationships are meaningful and can be used for course design and adaptive learning purposes. Additionally, the proposed approach can use student interactions with one *auxiliary* learning resource (such as examples) to predict students performance on another *target* learning resource type (such as problems). Our experiments on four real-world datasets show that our approach can efficiently use the extra information provided by auxiliary learning resources and significantly improve the student performance prediction error over the baseline models.

2. THE APPROACH

Our proposed approach is inspired by Canonical Correlation Analysis (CCA) [Hotelling 1936], which is a multivariate statistical model that studies the interrelationships among sets of multiple dependent and independent variables. CCA's goal is to find linear projections of these variable sets into a shared latent space such that the correlation between these projections are maximized. In this research, we use CCA as our main tool: we propose to find the relationship between students' ungraded activities (as independent variables) and students' graded activities (as dependent variables) using CCA. Our final goal is to propose a model for predicting student performance using pairs of resource types, motivated by the discovered relationships.

Our reason for choosing CCA as inspiration is twofold. First, CCA provides different views to the same data samples. Since we have the same students interacting with multiple resource types (e.g., examples and problems), we need to have a tool to model these interactions at the same time, while distinguishing between distinct resource types (as different views). Other factor analysis models, such as Principal Component Analysis (PCA), operate on one single view of the data and are not appropriate for our problem. Second, because of having multiple learning resources within each resource type (e.g., multiple problems and multiple examples) and several students (as datapoints) we need a multi-variate statistical model to capture the two-dimensional variability in the data. Bivariate or simpler multivariate models such as correlation or regression analysis can only capture the data variance for one dependent variable at a time and thus miss the variability of either students or learning material. We first give a brief background on CCA and then explain how to model and solve our problems using it.

CCA. If matrix $X_{m\times n}$ represents n data samples and m variables and matrix $Y_{p\times n}$ contains the values for p variables of same n data samples, CCA aims to find linear transformations, w_x and w_y, such that the correlation between projections of X and Y through w_x and w_y (reflected as ρ in Equation 1) is maximized.

$$\rho = \frac{w_x^T XY^T w_y}{\sqrt{(w_x^T XX^T w_x)(w_y^T YY^T w_y)}} \tag{1}$$

Since multiplication of w_x and w_y by a constant does not change the value of ρ in Equation 1, the problem of finding w_x and w_y can be formulated as in Equation 2.

$$\begin{aligned} &\max_{w_x, w_y} w_x^T XY^T w_y \\ &\text{subject to } w_x^T XX^T w_x = 1, w_y^T YY^T w_y = 1 \end{aligned} \tag{2}$$

Adding the regularization parameters to Equation 2, for controlling over-fitting of ρ, Sun et al. show that this regularized-CCA problem can be represented as in Equation 3, and solved using a least squares approach [Sun et al. 2008]. The formulation for w_y is a symmetrical version of Equation 3.

$$XY^T(YY^T)^{-1}YX^T w_x = \eta(XX^T + \lambda I)w_x \tag{3}$$

In addition to w_x and w_y that produce the maximum correlation ρ, there can be other projection vector pairs that can map X and Y matrices with correlations less than or equal to ρ. The optimization problem in Equation 4 finds these multiple projection vectors for X (in matrix W_x).

$$\begin{aligned}&\max_{W_x} \text{Trace}(W_x^T XY^T(YY^T)^{-1}YX^TW_x)\\&\text{subject to } W_x^TXX^TW_x = I\end{aligned} \tag{4}$$

2.1 Relation Discovery between Learning Resource Types

As students work with various learning resources that are provided in an online course or a tutoring system, they gain more knowledge about the concepts presented in the course and can tackle more complicated problems. Knowing the relationship between different learning resource types and the way they interact in affecting students' knowledge can help better course design.Having the learning material from one resource type (e.g., problems) as one set of variables and learning material from another type (e.g., examples) as the other set of variables, we can interpret canonical correlation as a measure of relatedness between resource types.

More specifically, to map our problem to the CCA setting, we suppose that there are n students that have at least one activity in each of the resource types. For example, these students may have tried some problems and studied some examples in the course. We represent the students' performance on problems as a matrix $Y_{p\times n}$, with n students, p problems, and $Y_{i,j}$ representing the student j's score in quiz i. This score can be a grade or pass/fail indicator. Similarly, students' example activities can be represented as another matrix $X_{m\times n}$, with n students, m examples, and $X_{i,j}$ as an indication that user j has read example i. Given these two activity matrices, we use CCA to find linear transformations W_x and W_y and canonical correlations P as in Equation 4.

Formulating our problem as an instance of CCA, W_x and W_y can represent linear transformation matrices that map the original activity matrices X and Y into a shared latent space. These projections are scaled based on the canonical correlation values in a diagonal matrix $P_{c\times c}$, in which each of the diagonal elements are equivalent to the canonical correlation value ρ_i for each projection vector pair $W_{x_{:,i}}$ and $W_{y_{:,i}}$. Meanwhile, the projection matrices $W_{x_{m\times c}}$ and $W_{y_{p\times c}}$ are representations of learning resources, projected into the shared space. Having this shared component space, we can compare and relate activities that are present in the two resource types.

In other words, each learning material i from the auxiliary learning resource in matrix X, will be represented as a $1\times c$ vector $W_{x_{i,:}}$ and each learning material j from the target learning resource in matrix Y, will be represented as a $1 \times c$ vector $W_{y_{j,:}}$. So, we can find the most similar resources from different types by looking at the cosine similarity between those vectors in the shared component space.

Note that this is different from simply comparing matrices X and Y in the shared *student* space by calculating their cosine similarity. Here, we have the canonical correlation effect on finding similar learning resources. To be more clear, if we suppose that $w_x^TXX^Tw_x = 1$ and $w_y^TYY^Tw_y = 1$ (by which we transformed Equation 1 to Equation 2), then we have:

$$\hat{\rho} = w_x^TXY^Tw_y \tag{5}$$

$\hat{\rho}$ in Equation 5 is equivalent to ρ in Equation 1, scaled by its denominator. Now, if we left-multiply both sides of Equation 5 by $w_x^{T^{-1}}$, and right-multiply both sides of it by ${w_y}^{-1}$, we achieve $XY^T = w_x^{T^{-1}}\rho {w_y}^{-1}$. Equivalently, when having multiple canonical correlations, we can see that:

$$XY^T = W_x^{T^{-1}}P{W_y}^{-1} \tag{6}$$

Equation 6 shows the relationship between the projection matrices with the cosine similarity of X and Y (XY^T). Clearly, YX^T and $W_yW_x^T$ are not equal.

2.2 Inter-Activity Performance Prediction

Predicting how a student performs on a problem can help teachers to adjust the course material based on students' predicted performance and can lead to personalized learning. Also, it can guide students towards a structured and effective learning. As in many prediction problems, educational data is usually incomplete: not all students try all resources. We focus on predicting students' scores for the first time that they try a problem. Thus, the problem of predicting students' performance can be interpreted as estimating the missing values in the student activity matrix (Y) that is described in the beginning of Section 2.

As proposed in Section 2.1, we can find the relationship between sets of learning resources of two types using CCA. Thus, if we know students' performance on auxiliary learning resources in matrix X and their performance in the target learning resource in matrix Y, we can understand how students' activities on auxiliary learning resources affect the same students' performance on the target learning resources. When the student activity matrix (Y) is incomplete, we can estimate w_x and w_y by calculating the canonical correlations between the auxiliary activity matrix X and the incomplete target activity matrix Y to achieve the estimated projection vectors $\hat{w_x}$ and $\hat{w_y}$. Using these projection vectors, we can estimate a complete activity matrix $\hat{Y}$ as in Equation 7.

$$\hat{Y} = \hat{w_y}\rho\hat{w_x}^TX \tag{7}$$

Here, student activities in the auxiliary learning resource are mapped to the shared latent space, scaled by the canonical correlation factor ρ, and then mapped back to the target learning resource space. In case of calculating multiple (c) projection vector pairs ($\hat{W}_{x_{m\times c}}$ and $\hat{W}_{y_{p\times c}}$), with canonical correlations represented in $P_{c\times c}$, we estimate students' performance ($\hat{Y}$) as in Equation 8.

$$\hat{Y} = \hat{W}_yP\hat{W}_x^TX \tag{8}$$

3. DATASETS

We use four datasets from two online platforms for our experiments. The anonymized data represent log files of student interaction with course resources (activities), and their performance in them. Each of these platforms allow their students to learn from multiple learning resource types that calls for modeling inter-activity relations. The first two datasets are richer since they have learning resource names, topics, and contents although we do not use them for the discovery and prediction purposes. The third and fourth datasets are larger, from a MOOC platform, with more variation in learning resource types. However, we do not have access to these learning resources beyond their assigned IDs. In the following sections, we describe each of these datasets.

Table 1: Statistics of Mastery Grids datasets

		students	prob.	Parsons prob.	annot. exam.	anim. exam.
Python	number	319	37	43	58	53
	average activity records	65.5	147.5	112.3	97.2	93.8
	density	0.34	0.46	0.35	0.30	0.29
Java	number	206	113	-	101	50
	average activity records	127.2	108.3	-	93.9	89.7
	density	0.78	0.53	-	0.47	0.44

Table 2: Statistics of Canvas Network datasets

		students	quiz-assign.	assign.	discus. topics
Business and Management	number	232	32	38	34
	average activity records	62.7	208.1	190.8	18.9
	density	0.60	0.89	0.82	0.08
Professions and Applied Sciences	number	1160	18	26	70
	average activity records	16.25	427.3	372.5	21.1
	density	0.14	0.37	0.32	0.02

3.1 Mastery Grids Datasets

Our first two datasets are collected from an online intelligent tutoring system, Mastery Grids [Loboda et al. 2014]. This system provides personalized access to three types of interactive content for Java programming and four types of content for Python programming. Parameterized semantic problems, annotated examples (code snippets with explanations), and animated examples (interactive simulations that visually demonstrate the runtime behavior of a code snippet) are the three types of resources that are available for both Java and Python courses. In addition to those, Python course includes the so-called *Parsons problems* originally introduced in [Parsons and Haden 2006].

The parameterized semantic problems (problems, for short) are generated by QuizJet and QuizPet system [Hsiao et al. 2009] from a pool of parameterized questions on Java and Python programming. As a result, the same problem can be attempted multiple times by students with various parameters. We only consider students' first attempt on each problem for our experiments. Annotated examples presented by WebEx allow students to interactively explore line-by-line explanation of code snippets [Brusilovsky and Yudelson 2008]. Working with animated examples, which are generated using Jsvee library [Sirkiä 2016], students can execute a Java or Python program visually, observing internal operation, such as variable assignments and printing on a console. In Parsons problems, students are asked to solve a programming task by selecting and sorting provided code lines.

Mastery Grids groups different learning resources into multiple learning topics. Although this system offers a recommended topic sequence in its interface, the students are free to select and work on any of the topics and learning resources at any given time. The Java dataset from this system is collected from Fall and Spring semesters of 2016. Among all of the students, we selected the ones who have at least one activity in each of the problems, annotated examples, and animated examples. A summary of statistics for these datasets are shown in Table 1. The Python dataset about two times sparser than the Java dataset in terms of number of all activities per student. Among different resource types, the density of student activities on problems are the closest between the two datasets. In both of the datasets, student activities on problems are the densest and activities on animated examples are the most sparse.

3.2 Canvas Network Datasets

Our third and fourth datasets are publicly available from Canvas Network (`http://canvas.net`) [Network 2016]. Canvas Network hosts many freely available open online courses in which it offers multiple leaning resource types. More specifically, in addition to learning modules, each course can have different types of assignments, discussions, and pop-quizzes. Participants are not limited to a specific sequence of learning material or assignments. Categories of the learning resources include "assignments", "quiz-assignments", "pop-quizzes", "discussions", and "wikis". The dataset is anonymized such that student IDs, course names, discussion contents, submission contents, and course contents are not available.

Course assignments can be quiz-style ("quiz-assignment") or in longer format, for which students submit a text or video file ("assignments"). We choose two of the offered courses in Canvas Network as the third and fourth datasets for our experiments. These two courses are selected because they provide multiple learning resource types and have more active students in all of these resource types. The first course is in the "Professions and Applied Sciences" field and the second course is in the "Business and Management" field.

Since assignments, quiz-assignments, and discussions have the most activities, we focus on these resource types in our experiments. Among these three, assignments and quiz-assignments are graded. For consistency, we normalize students' grades between zero and one based on their maximum possible grade. For discussions, we consider a binary variable representing if a student has posted a message or not. We select the students who have at least one activity in each of these learning resources. A summary of statistics for these datasets is shown in Table 2. Discussion topics have the least dense activity matrices in the two datasets. They are very sparse compared to student activities on assignments and quiz-assignments. Comparing the two datasets from Canvas Network, overall student activities in professional and applied sciences domain course is very sparse. But, the density of student activities on all resources in business and management domain course is comparable with the datasets from Mastery Grids system. However, the distribution of student activities among various resource types are more skewed in the Canvas Network datasets.

4. EXPERIMENTS

4.1 Experiment Setup

Per the proposed model in Section 2, element $X_{i,j}$ in activity matrix X represents the result of student j's first attempt on learning resource i. This activity result can be different for different learning resource types. For graded learning resources, such as assignments and quiz-assignments, we use the normalized score of students; for problems and Parsons problems with success or failure feedback, we use binary scores; and for non-graded activities, such as reading an annotated example or posting in a discussion forum, we use a binary indicator that shows the students' attempt. We use average imputation for missing values.

For prediction experiments, we follow a 5-fold user stratified separation of the student performance data to perform cross-validation on it. Particularly, in each round of experiments, we select 20% of students as test students, 15% of them for validation purposes, and 65% of them as train. Our task is to predict test students' performance on activities in a target learning resource type, observing 20% of these students' activities, and the training data. In the CCA-based proposed approach, the training data includes all students' activities in the auxiliary learning resource type, in addition to observed activities of students in the target resources. We repeat each round of the experiments for 5 times.

Since only quiz-assignments and assignments are graded in the Canvas Network datasets, and only problems and Parsons problems are graded in the Mastery Grids datasets, we define the prediction tasks on these resource types. Discussions from the Canvas Network datasets and examples (annotated and animated) from the Mastery Grids datasets are only used as auxiliary resources. Note that each of graded resource types (quiz-assignments, assignments, problems, and Parsons problems) can also be used as an auxiliary resource for another type of graded resource in the same dataset.

Baselines. In previous works, collaborative filtering methods have been proved successful in predicting students performance [Thai-Nghe et al. 2011, Sahebi et al. 2014]. Since our proposed approach is similar to these approaches in discovering latent relationships among learning resources, through factorizing activity matrices, we choose two settings of SVD++ algorithm [Koren et al. 2009] as our baselines. To study if adding student activities in auxiliary resource type would help better estimation of students performance in the target resource type, we compare our approach with single-resource SVD++ algorithm. In this setting we run SVD++ algorithm only on the target learning resource matrix, assuming that we do not have the information on student activities in the auxiliary resource types, and compare the results with our proposed method. To understand our CCA-based method's efficiency on capturing important relationships between different learning resource types, we compare it with a paired-resource setting of SVD++ algorithm. Particularly, we merge the two auxiliary and target learning resource types into one set of learning materials (represented by one matrix) and run the SVD++ algorithm on this augmented matrix. Note that our proposed method factorized two separate matrices at the same time but SVD++ can only factorize one matrix.

Since the student activity datasets are biased towards student success (e.g., average grade for problems in the Python dataset is 0.67 out of 1), we compare the methods with an average baseline. To do this, we use the training dataset average as the predicted performance for all of the students in each of the 5 data splits.

4.2 Discovering Relationships between Learning Resource Types

One of our goals in this paper is to understand relationships and interactions between sets of learning resources with various types. CCA has the ability to represent each pair of learning resource types in the same latent space. This enables us to relate learning material of different types only based on student activities, without relying on their content or presented concepts. Since the Mastery Grid datasets provide learning resource names and topics we can confirm the discovered relationships by comparing them with learning resource topic similarities. These topics have been manually assigned to learning resources by experts, during course design in Mastery Grids. In order to take a deeper look at the discovered similarities, we study the top similar learning resources of different types in the same course (as shown in Table 3). To calculate these similarities, we look at projections of each learning resource in the shared latent space, W_x and W_y and calculate the cosine similarity between them, as mentioned in Section 2.1. We look at the most similar learning resources of each course in the following.

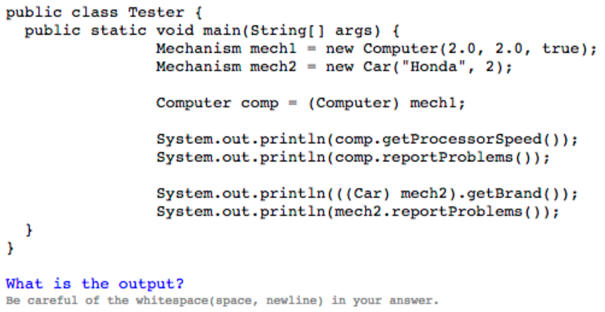
The Java Dataset. For the Java dataset, we can calculate the cosine similarity of problems with animated examples and problems with annotated examples. We can see the most similar problems and animated examples in rows 1-4 of Table 3. As we can see, three of these four learning resource pairs are from the same expert-labeled topic. For example, both problem “jWhile1” and animated example “ae_while_demo” are about “while loops” in Java. This shows that our approach can accurately figure out the most similar problems and animated examples, only based on student activities and their performance, not knowing about their topic or content. However, the resources in row 3 are from different expert-labeled topics “boolean expressions” and “switch”. While these two are not exactly the same, the switch expressions in Java use boolean expressions in their conditional statements. So these two topics are closely related to each other: if a student cannot understand the “boolean expressions” topic, understanding the “switch” topic would be difficult for this student.

The most similar Java annotated examples and problems, found by CCA projection matrices, are listed in rows 5-8. Here, we do not see the obvious similarities that was apparent between animated examples and problems. In row 5, there is topic similarity between the problem with “loops do-while” topic and the annotated example with “loops for” topic: both of them are about loops in Java. For row 8, we know that Java for loops use “arithmetic operations” in their conditional statement. However, topics for similar resources discovered in rows 6 and 7 look irrelevant. Row 6's problem is labeled by experts with the “interfaces” topic, while the similar annotated example is labeled with the “variables” topic. Likewise, the problem topic in row 7 is “interfaces”, while the topic of similar annotated example is “objects”.

To gain more insight about these learning resources, we looked at their contents. We discovered that although the general topics for these problems and their discovered anno-

Table 3: Most similar learning materials of different types, from Java and Python courses, according to their similarity using CCA projection vectors.

course	*material type*	*row*						
Java	*prob. & anim. exam.*		*prob. ID*	*prob. name*	*prob. topic*	*anim. exam. topic*	*anim. exam. name*	*anim. exam. ID*
		1	14	jArrayList5	ArrayList	ArrayList	ae_arraylist2_v2	3
		2	18	jBoolean_Operators	Boolean expressions	Switch	ae_switch_demo2	44
		3	65	jMathFuc2	Arithmetic operations	Arithmetic operations	ae_arithmetic_v2	1
		4	100	jWhile1	Loops while	Loops while	ae_while_demo	49
	prob. & annot. exam.		*prob. ID*	*prob. name*	*prob. topic*	*annot. exam. topic*	*annot. exam. name*	*annot. exam. ID*
		5	37	jDowhile1	Loops do_while	Loops for	for1_v2	28
		6	57	jInterfaces1	Interfaces	Variables	PrintTester	78
		7	61	jInterfaces5	Interfaces	Objects	AccessorMutatorDemo	1
		8	63	jMathCeil	Arithmetic operations	Loops for	JavaTutorial_4_6_8	57
Python	*prob. & annot. exam.*		*prob. ID*	*prob. name*	*prob. topic*	*annot. exam. topic*	*annot. exam. name*	*annot. exam. ID*
		9	3	q_py_arithmetic1	Variables	Variables	pyt1.3	5
		10	21	q_py_nested_if_elif1	if_statements	values_references	pytt10.25	58
		11	23	q_py_obj_account1	classes_objects	Lists	pyt7.2	53
	prob. & anim. exam.		*prob. ID*	*prob. name*	*prob. topic*	*anim. exam. topic*	*anim. exam. name*	*anim. exam. ID*
		12	7	q_py_dict_access1	dictionary	loops	ae_adl_while	39
		13	29	q_py_output1	output_formatting	variables	ae_adl_arithmetics2	1
		14	10	q_py_fun_car1	functions	exceptions	ae_adl_tryexcept2	34
	prob. & pars. prob.		*prob. ID*	*prob. name*	*prob. topic*	*pars. prob. topic*	*pars. prob. name*	*pars. prob. ID*
		15	10	q_py_fun_car1	functions	exceptions	ps_python_try_adding	38
		16	12	q_py_if_elif1	if_statements	loops	combo_python_while	9
		17	35	q_py_swap1	variables	variables	combo_swap	11
	pars. prob. & annot. exam.		*pars. prob. ID*	*pars. prob. name*	*pars. prob. topic*	*annot. exam. topic*	*annot. exam. name*	*annot. exam. ID*
		18	1	combo_avg	variables	variables	pyt2.1	32
		19	14	ps_python_addition	variables	variables	pyt1.2	4
		20	41	ps_return_bigger_or_none	functions	functions	pyt10.7	30
	pars. prob. & anim. exam.		*pars. prob. ID*	*pars. prob. name*	*pars. prob. topic*	*anim. exam. topic*	*anim. exam. name*	*anim. exam. ID*
		21	1	combo_avg	variables	variables	ae_python_assignment	40
		22	12	ps_hello	variables	variables	ae_adl_arithmetics2	1
		23	43	ps_simple_params	variables	functions	ae_adl_returnvalue	29

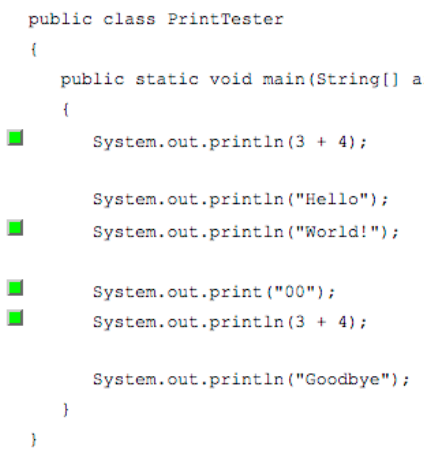


```
public class Tester {
  public static void main(String[] args) {
              Mechanism mech1 = new Computer(2.0, 2.0, true);
              Mechanism mech2 = new Car("Honda", 2);

              Computer comp = (Computer) mech1;

              System.out.println(comp.getProcessorSpeed());
              System.out.println(comp.reportProblems());

              System.out.println(((Car) mech2).getBrand());
              System.out.println(mech2.reportProblems());
  }
}

What is the output?
Be careful of the whitespace(space, newline) in your answer.
```

Figure 1: Content of problem with "Interfaces" topic (row 6 of Table 3)

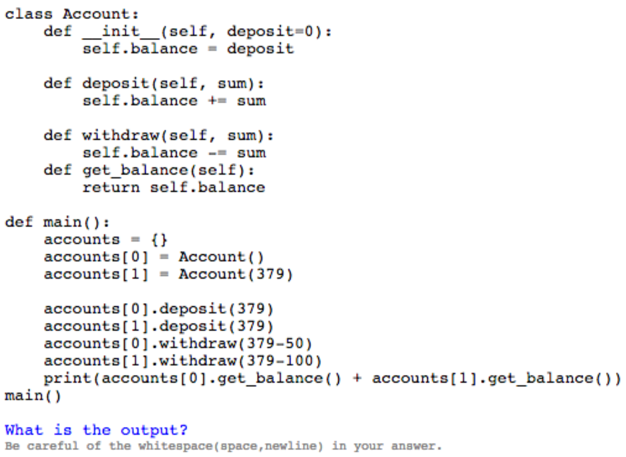
tated examples are not the same, they include very similar concepts. For example, Figure 1 shows the content for problem "jInterfaces1" (topic: "interfaces"), and Figure 2 shows the content for annotated example "PrintTester" (topic: "variables"). As we can see, the concept of printing an output in the console is very important in both of these learning resources. Interestingly, it appears that although the designers of Java course were interested in the mentioned topics while designing these learning resources, we are discovering other possible "latent topics" for them. Another factor in these newly-found relations can be the mixed relationship of annotated examples with students performance. Hosseini et al. have studied the use and impact of annotated and animated examples in the same online tutoring system and concluded that students are likely to learn more from animated examples [Hosseini et al. 2016]. Particularly, they showed that although more views of animated examples is associated with a higher course grade, the number of views on annotated examples has a negative effect on it. A possible reason is the negative process of associating examples with poor knowledge: students with poor knowledge are more likely to study annotated examples. This association can potentially overcome the positive impact of learning from annotated examples and lead to a negative impact. Also, they show that animated examples provided better impact on problem solving success and post-test scores.

The Python Dataset. We study 5 pairs of resource types and the cosine similarities between W_ys and W_x^Ts in the Python dataset: problems vs. animated examples, problems vs. annotated examples, Parsons problems vs. animated



```
public class PrintTester
{
   public static void main(String[] args)
   {
      System.out.println(3 + 4);

      System.out.println("Hello");
      System.out.println("World!");

      System.out.print("00");
      System.out.println(3 + 4);

      System.out.println("Goodbye");
   }
}
```

Figure 2: Content of annotated example with "Variables" topic (row 6 of Table 3)



```
class Account:
    def __init__(self, deposit=0):
        self.balance = deposit

    def deposit(self, sum):
        self.balance += sum

    def withdraw(self, sum):
        self.balance -= sum
    def get_balance(self):
        return self.balance

def main():
    accounts = {}
    accounts[0] = Account()
    accounts[1] = Account(379)

    accounts[0].deposit(379)
    accounts[1].deposit(379)
    accounts[0].withdraw(379-50)
    accounts[1].withdraw(379-100)
    print(accounts[0].get_balance() + accounts[1].get_balance())
main()
```

What is the output?
Be careful of the whitespace(space,newline) in your answer.

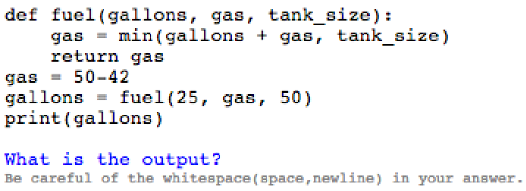
Figure 3: Content of problem with "classes_objects" topic (row 11 of Table 3)

examples, Parsons problems vs. annotated examples, and problems vs. Parsons problems. Samples of discovered similar learning resources are shown in Table 3.

As shown in rows 9-11, the first problem and its matched annotated example have the same topic of "variables". But, the next two pairs do not have a common topic. We study the content of these learning resources to understand the nature of their similarity. For example, if we look at row 11, we see that annotated example "pyt7.2" has topic of "lists". Now if we look at problem "q_py_obj_account1" with topic of "classes_objects" in Figure 3, we can see that this problem uses lists (accounts variable) in it. We avoid showing the content for the pair in row 10 due to space limits.

Rows 12-14 show similar animated examples and problems in the Python dataset. To show the similarities between concepts used in these animated examples and problems, we look at one pair: problem "q_py_fun_car1" with topic "functions" (Figure 4) and animated example "ae_adl_tryexcept2" with topic "exceptions" (Figure 5). We can see that there is a function call and a function definition in this animated example (Figure 5). Consequently, although this animated example is not designed to teach the "function" topic and despite of it being labeled with the "exceptions" topic only, the discovered similarities show the associations between students' learning of functions and this animated example.

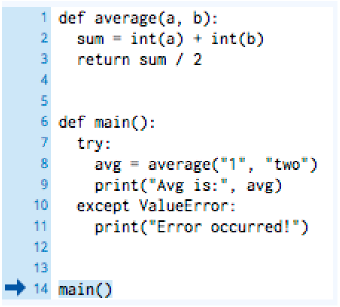
The most similar problems and Parsons problems are shown in rows 15-17 of Table 3. Two of the top similar pairs are from the same ("variables") or related ("if statements" and "loops") topics. The resources in row 15 are from different topics: a "functions" problem and an "exceptions" Parsons problem. But, as can be seen in Figures 4 and 6 the Parsons problem includes a function definition. So, students can learn about functions while executing this animated example that is about exceptions.



```
def fuel(gallons, gas, tank_size):
    gas = min(gallons + gas, tank_size)
    return gas
gas = 50-42
gallons = fuel(25, gas, 50)
print(gallons)
```

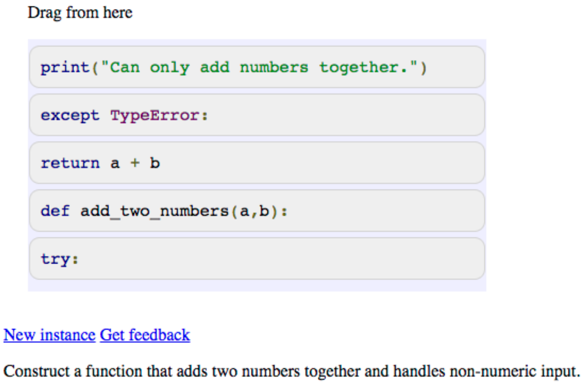
What is the output?
Be careful of the whitespace(space,newline) in your answer.

Figure 4: Content of problem with "functions" topic (rows 14 and 15 of Table 3)



```
def average(a, b):
  sum = int(a) + int(b)
  return sum / 2


def main():
  try:
    avg = average("1", "two")
    print("Avg is:", avg)
  except ValueError:
    print("Error occurred!")


main()
```

Figure 5: Content of animated example with "exceptions" topic (row 14 of Table 3)



Drag from here

```
print("Can only add numbers together.")
```
```
except TypeError:
```
```
return a + b
```
```
def add_two_numbers(a,b):
```
```
try:
```

New instance Get feedback

Construct a function that adds two numbers together and handles non-numeric input.

Figure 6: Content of Parsons problem with "Exceptions" topic (row 15 of Table 3)

Finally, as we can see in rows 18-23, analogous samples of Parsons problems vs annotated examples, and Parsons problems vs animated examples are all from the same topics.

One may think that the discovered similarities are a result of topic arrangements in the course design and conclude that we can find these similar learning resources by only looking at the co-occurrence of student activities in two learning resource types, e.g., by calculating the cosine similarities between learning resources in the original student-space, or matrices X and Y. However, looking at some of the discovered similarities, such as the second row of Table 3, reassures us that our approach can find the relationships beyond their trivial co-occurrence. As we have mentioned, the "switch" and "boolean Expressions" topics are not the same, but are very related. In the Mastery Grids interface, these two topics are not placed right next to each other. But, another topic ("if-else" topic) is placed between them. This means that the discovered similarity is not solely based on activity co-occurrence due to topic placement in Mastery Grids.

To discover what we can gain from trivial co-occurrences, without using our proposed method, we looked at samples of the most similar learning resources, based on the cosine similarity between student activities in the original student space (similarity between matrices X and Y). In this case, the most similar discovered learning resource pairs are either placed closely in the same topic (and thus, may happen due to the students following the sequence imposed by learning resource arrangements in the interface), or do not have any meaningful content-based relationship. For example, the most similar animated example that is discovered in student space for the "jBoolean_Operators" problem (problem in row 2 of Table 3) is labeled with the "primitive data types" topic, demonstrating "Double" and "Short" data types.

To summarize, the discovered CCA-based similarities in both datasets are meaningful. Some of the related learning resource pairs are from the same topics, others are related in the concepts or sub-topics that they present. In general, this is a very promising result, especially for applications in which the learning resource contents are difficult to analyze and compare. Discovering these similarities, instructors can rearrange their learning material in ways that most benefits students' learning. Also, it can be used for multi-source knowledge modeling of students. Namely, we can model student knowledge in shared concepts between problems and animated examples and understand how a student's ability in a learning recourse type (e.g., to solve a problem) increases by trying another learning resource of a different type (e.g., a related animated example).

4.3 Predicting Student Performance Using Auxiliary Resource Types

Using the formulation proposed in Section 2.2, our goal here is to predict students' performance using auxiliary learning resource types and compare it with similar baseline approaches. We measure performance of the proposed and baseline approaches using Root Mean Squared Error (RMSE). This measure quantifies the average difference between actual students' score and their predicted performance.

Mastery Grids Datasets For the Java programming dataset, we run two sets of experiments. The first set of experiments is on predicting students performance on problems, using their activities on annotated examples as auxiliary data ("annotated examples → problems"). In the second set of experiments, we use animated example activities as the auxiliary resource for predicting students performance on problems ("animated examples → problems"). As mentioned before, we compare the results of our proposed approach with single-resource SVD++ –only using student logs on problems– and paired-resource SVD++ –with the same input as our proposed approach–.

For the Python programming dataset, we run six sets of experiments. Having problems and Parsons problems as target learning resource types, we use annotated examples and animated examples as the auxiliary learning resources. Additionally, problems may help us in predicting students' performance in Parsons problems, and vice versa.

Table 4 shows the RMSE of CCA-based and baseline approaches for these sets of experiments on both of Mastery Grids datasets. The numbers in parentheses report the 95-percentile confidence interval for the reported errors. As we can see here, our CCA-based approach performs significantly better than the baselines in all of the experiment setups in both datasets. As our proposed approach performs better than single-resource SVD++, we can conclude than adding the auxiliary data significantly improves student performance prediction. On the other hand, we can see that the proposed CCA-based approach works better than SVD++ in the multi-recourse setting using the same set of auxiliary and target data. Therefore, we can conclude that our approach is a better fit for effectively using auxiliary data.

Comparing the two settings for SVD++, in the Python dataset single-resource SVD++ performs as good as, or significantly better than paired-resource SVD++. Specifically, for combinations "animated examples → problems" and "annotated examples → problems", paired-resource SVD++ has a significantly higher error than single-resource SVD++. This confirms our findings in Section 4.2 about smaller similarities between problems and examples in the Python dataset. As expected in biased datasets, we can see that average baseline is working very well. Comparing with paired-resource SVD++, its error is significantly lower in four of the experiments on the Python dataset. Single-resource SVD++ is significantly better than (in "animated examples → problems", "annotated examples → problems", and "problems → Parsons problems") or similar to the average baseline.

In contrast, in the Java dataset, the average baseline has slightly, but significantly, higher error than the proposed approach and the other two baselines for "annotated examples → problems". For "animated examples → problems", the average baseline has better predictions compared to the other two baselines. Also, paired-resource SVD++ works significantly better than single-resource SVD++ for "annotated examples → problems". This shows that paired-resource SVD++ is not consistent on different datasets, even if similar learning resource types are used, and to be able to take advantage of auxiliary information, a more advanced approach, such as the proposed one, is needed.

Canvas Network datasets. Canvas Network datasets give us the opportunity to test our approach on more varied data of MOOCs and in different domains. Notably, "Professions and Applied Sciences" data has more users and is very

Table 4: RMSE for student performance prediction task on Mastery Grids datasets.

		anim. example → problem	annot. example → problem	pars. prob. → problem	anim. example → pars. prob.	annot. example → pars. prob.	prob. → pars. prob.
Java	paired-resource CCA	**0.4148 (0.0097)**	**0.4159 (0.0057)**	-	-	-	-
	paired-resource SVD++	0.5304 (0.0127)	0.4696 (0.0047)	-	-	-	-
	single-resource SVD++	0.5178 (0.0214)	0.4537 (0.0119)	-	-	-	-
	average baseline	0.4859 (0.0071)	0.4854 (0.0039)	-	-	-	-
Python	paired-resource CCA	**0.4584 (0.0035)**	**0.4566 (0.0024)**	**0.4579 (0.007)**	**0.4122 (0.0081)**	**0.4098 (0.0043)**	**0.4105 (0.0075)**
	paired-resource SVD++	0.516 (0.0124)	0.5122 (0.0156)	0.5524 (0.0083)	0.5213 (0.022)	0.456 (0.0084)	0.4954 (0.0123)
	single-resource SVD++	0.4921 (0.0147)	0.4921 (0.0147)	0.4921 (0.0147)	0.4409 (0.0059)	0.4409 (0.0059)	0.4409 (0.0059)
	average baseline	0.4961 (0.0024)	0.4972 (0.0036)	0.4957 (0.0014)	0.4724 (0.0056)	0.4716 (0.0047)	0.4723 (0.0072)

Table 5: RMSE for student performance prediction task on Canvas Network datasets, using discussions, quiz-assignments, and assignments as auxiliary resources.

		quiz-assignments → assignments	discussions → assignments	assignments → quiz-assignments	discussions → quiz-assignments
Business and Management	paired-resource CCA-based	**0.1073 (0.0209)**	**0.1093 (0.0163)**	**0.0911 (0.0124)**	**0.1207 (0.0109)**
	paired-resource SVD++	0.1871 (0.0143)	0.1569 (0.0115)	0.1696 (0.0111)	0.1903 (0.0085)
	single-resource SVD++	0.1890 (0.0208)	0.1890 (0.0208)	0.1532 (0.0125)	0.1532 (0.0125)
	average baseline	0.1741 (0.0182)	0.1741 (0.0182)	0.1752 (0.0118)	0.1752 (0.0118)
Professions and Applied Sciences	paired-resource CCA-based	**0.1264 (0.0085)**	**0.1252 (0.0049)**	**0.1252 (0.0035)**	**0.1287 (0.0105)**
	paired-resource SVD++	0.2070 (0.0112)	0.1897 (0.0140)	0.2039 (0.0211)	0.3254 (0.0171)
	single-resource SVD++	0.5235 (0.0196)	0.5235 (0.01960)	0.2057 (0.0176)	0.2057 (0.0176)
	average baseline	0.4596 (0.0019)	0.4596 (0.0019)	0.3838 (0.0037)	0.3838 (0.0037)

sparse compared to all other datasets. For Canvas Network datasets we run four sets of experiments. In the first two sets, we use quiz-assignments and discussions as auxiliary resources to predict students' performance in assignments. In the third and fourth sets of experiments we predict students' grade in quiz-assignments using general assignments and discussions as auxiliary resources.

Table 5 shows RMSE of all approaches on both "Professions and Applied Sciences" and "Business and Management" datasets. Similar to our results on the Mastery Grids dataset, we can see that the proposed approach can effectively use auxiliary resources to provide better estimation of student performance in all resource pairs. Comparing paired-resource SVD++ to single-resource SVD++, we can see that in most of the experiments their error is not significantly different. Only for "quiz-assignments → assignments" and "discussions → assignments", in "Professions and Applied Sciences" dataset, paired-resource SVD++ is significantly better than single-resource SVD++. Comparing the average baseline results, it's error is significantly higher than (in "Professions and Applied Sciences" dataset) or similar to paired-resource SVD++. Whereas compared to single-resource SVD++, it works better in predicting assignments, and worse in predicting quiz-assignments. This is because there is more variation in students' scores in quiz-assignments.

In addition to the way different courses are designed and learning resources are prepared, one of the reasons behind the different results between the two datasets can be due to the variations between two course datasets. For example, having more students and being sparser may lead to added value of auxiliary information in the "Professions and Applied Sciences" dataset (Table 2). In other words, effectiveness of adding auxiliary data for the task of performance prediction depends on the dataset and its characteristics.

5. CONCLUSIONS

We proposed an approach inspired by canonical correlation analysis for discovering interrelationships between learning resources of different types, only using student performance in them. This approach can also be used to predict students' performance. That is to say, we can predict students' performance in one type of learning resources, with the help of student activities in another resource type. We evaluated the proposed approach with four datasets and two tasks.

For the task of finding learning resource interrelationships, we evaluated our approach on the Java programming dataset with three resource types, and the Python programming dataset with four resource types. Finding the most similar resources of different types, only based on student activities, we showed that our approach is very promising in detecting these similarities, especially for learning resources that have been proved to have a positive effect on students' learning. Also, we found that our approach goes beyond the designated topics for learning resources and discovers latent similarities that provide clues of their content similarity.

Having four datasets from two online learning systems, we ended up with 16 total experiment sets for predicting student performance in paired resource types. We compared our proposed approach with an average baseline and two algorithmic baselines: one using student activities in both auxiliary and target resource types (paired resource SVD++), and one with using student activities in only target resource type (single resource SVD++). The experiments showed that our proposed approach can significantly improve estimation of student grades in all setups and datasets. This success is in part due to the extra information from the auxiliary resource types on students' performance: in three out of 16 setups, the baseline algorithm with auxiliary data performed better than the baseline algorithm without auxiliary data . However, in two of the setups the baseline with auxiliary data performed significantly worse than the baseline without it. Meanwhile, the proposed approach performed better than both baselines in all of the 16 experiments. It showed that better performance of the proposed approach is not only because of having extra information, but also because of its ability to use latent interrelationships between auxiliary and target resource types, in a more efficient way.

6. REFERENCES

[Agrawal et al. 2014] R. Agrawal, M. Christoforaki, S. Gollapudi, A. Kannan, K. Kenthapadi, and A. Swaminathan. 2014. Mining videos from the web for electronic textbooks. In *International Conference on Formal Concept Analysis*. 219–234.

[Beck et al. 2008] J. Beck, K. Chang, J. Mostow, and A. Corbett. 2008. Does help help? Introducing the Bayesian Evaluation and Assessment methodology. In *Intelligent Tutoring Systems*. 383–394.

[Brusilovsky and Yudelson 2008] P. Brusilovsky and M. Yudelson. 2008. From WebEx to NavEx: Interactive Access to Annotated Program Examples. *IEEE* 96 (2008), 990–999.

[Hosseini et al. 2016] R. Hosseini, T. Sirkiä, J. Guerra, P. Brusilovsky, and L. Malmi. 2016. Animated Examples As Practice Content in a Java Programming Course. In *Technical Symposium on Computing Science Education*. 540–545.

[Hotelling 1936] H. Hotelling. 1936. Relations Between Two Sets of Variates. *Biometrika* 28, 3/4 (1936).

[Hsiao et al. 2009] I. Hsiao, S. Sosnovsky, and P. Brusilovsky. 2009. Adaptive navigation support for parameterized questions in object-oriented programming. In *European Conference on Technology Enhanced Learning*. 88–98.

[Huang et al. 2015] Y. Huang, J. P. González-Brenes, and P. Brusilovsky. 2015. Challenges of Using Observational Data to Determine the Importance of Example Usage. In *International Conference on Artificial Intelligence in Education*. 633–637.

[Jiří and Pelánek 2017] R. Jiří and R. Pelánek. 2017. Measuring Similarity of Educational Items Using Data on Learners' Performance. In *Educational Data Mining*. 16–23.

[Koren et al. 2009] Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[Loboda et al. 2014] T. D. Loboda, J. Guerra, R. Hosseini, and P. Brusilovsky. 2014. Mastery grids: An open source social educational progress visualization. In *European Conference on Technology Enhanced Learning*. 235–248.

[Najar et al. 2014] A. Najar, A. Mitrovic, and B. M. McLaren. 2014. Adaptive Support versus Alternating Worked Examples and Tutored Problems: Which Leads to Better Learning?. In *International Conference on User Modeling, Adaptation, and Personalization*. 171–182.

[Network 2016] Canvas Network. 2016. Canvas Network Courses, Activities, and Users (4/2014 - 9/2015) Restricted Dataset. (2016). `https://doi.org/10.7910/DVN/XB2TLU`

[Parsons and Haden 2006] D. Parsons and P. Haden. 2006. Parson's Programming Puzzles: A Fun and Effective Learning Tool for First Programming Courses. In *Australasian Conference on Computing Education*. 157–163.

[Sahebi et al. 2014] S. Sahebi, Y. Huang, and P. Brusilovsky. 2014. Parameterized Exercises in Java Programming: using Knowledge Structure for Performance Prediction. In *Workshop on AI-supported Education for Computer Science*. 61–70.

[Sao Pedro et al. 2013] M. Sao Pedro, R. Baker, and J. Gobert. 2013. Incorporating scaffolding and tutor context into bayesian knowledge tracing to predict inquiry skill acquisition. In *Educational Data Mining*.

[Sirkiä 2016] T. Sirkiä. 2016. Jsvee Kelmu: Creating and Tailoring Program Animations for Computing Education. In *Working Conference on Software Visualization*. 36–45.

[Sun et al. 2008] L. Sun, S. Ji, and J. Ye. 2008. A least squares formulation for canonical correlation analysis. In *International Conference on Machine Learning*. 1024–1031.

[Thai-Nghe et al. 2011] N. Thai-Nghe, L. Drumond, T. Horváth, A. Nanopoulos, and L. Schmidt-Thieme. 2011. Matrix and Tensor Factorization for Predicting Student Performance.. In *International Conference on Computer Supported Education*. 69–78.

[Velasquez et al. 2014] N. Velasquez, I. Goldin, T. Martin, and J. Maughan. 2014. Learning Aid Use Patterns and Their Impact on Exam Performance in Online Developmental Mathematics. In *Educational Data Mining 2014*. 379–380.

[Wen and Rosé 2014] M. Wen and C. P. Rosé. 2014. Identifying latent study habits by mining learner behavior patterns in massive open online courses. In *International Conference on Information and Knowledge Management*. 1983–1986.

How many friends can you make in a week?: evolving social relationships in MOOCs over time

Yiqiao Xu
Department of Computer Science
NCSU, Raleigh, NC, USA
yxu35@ncsu.edu

Collin F. Lynch
Department of Computer Science
NCSU, Raleigh, NC, USA
cflynch@ncsu.edu

Tiffany Barnes
Department of Computer Science
NCSU, Raleigh, NC, USA
tmbarnes@ncsu.edu

ABSTRACT

Massive Open Online Courses (MOOCs) are designed on the assumption that good students will help poor students thus offloading the individual support tasks from the instructor to the class. However prior research has shown that this is not always true. Students in MOOCs tend to form distinct sub-communities and their grades are closely correlated with those of their closest peers. That work, however, was only based on analyzing the final social network in a MOOC. In this paper, we study the evolution of these co-performing clusters over time. We explore a longitudinal approach to detect how students form their social connections on the discussion forum and we show that students form close coequal communities early in the course and maintain them over the duration of the course.

Keywords

MOOC, social network analysis, community detection, forum participation

1. INTRODUCTION

One promise of Massive Open Online Courses (MOOCs) is that we can provide high-quality educational content to students around the world at relatively low cost. The broad goal of MOOCs is to scale instruction by allowing expert instructors to provide guidance to hundreds or even thousands of students at a time. Such large-scale education has the potential to be revolutionary both for individual students and for educational systems. The current generation of MOOCs are designed to achieve this scaling by outsourcing much of the individual support tasks to students. That is, rather than capping enrollment to ensure that the instructor and TAs can support every students' needs, MOOCs provide online forums that encourage students to share common questions and to provide collaborative guidance or to benefit from each others' interactions with the limited support staff. Thus it is tacitly assumed that students will have common issues and that good students will help poor students with course content, assignments, logistics, and other issues. The role of instructors and TAs is then often to *curate* help rather than *authoring* it.

In a prior study Brown et al. examined the formation of communities in a large scale MOOC on Big Data in Education [3]. They extracted social networks from the online course forum and analyzed the connections between students. Contrary to the implicit assumption described above, they found that the social connections were not evenly distributed. Nor did they find that the lower-performing students made persistent connections with their higher-performing peers. Instead they found that the students formed distinct sub-communities and that their performance in the course was strongly correlated with that of their closest neighbors. In followup work, Brown et al. also found that these communities were not aligned with students' shared backgrounds nor were they apparently driven by shared course goals [2]. They further found that these results were stable even after the instructional staff and other highly-connected or *hub* students were factored out. Thus the authors concluded that the pattern of students' social relationships can be used to predict their performance and that interventions which target those social relationships may help students to improve either by selecting good peers or by flagging isolated and poorly-performing groups for individual attention.

That work, however, was limited by the fact that it only used the *final* social network from the course. Thus when evaluating students' performance the authors included all posts and social interactions that had developed over the duration of the course. In order to provide useful guidance during the course and to provide reliable information to instructors, we must show that it is possible to detect these relationships based upon partially-formed networks. In general most students' patterns of help-seeking change over the duration of the course. Students often drop out of courses, particularly MOOCs, or taper off their involvement as they lose interest. Students also face difficulties in courses that may make them scale up their communication as the course becomes more challenging. It may be the case that the network structure will change radically over the course of the class and that any early detection model or instructor dashboard will be erratic, invalid, or simply out of date.

In this work, we expand upon the prior work of Brown et al. by examining the growth of the students' social relationships over time, in the same MOOC. To that end we segmented the forum data by time and performed a sequential analysis of the evolving social network. Our goal in this work will be to address the following questions: First, are students' social groups stable over time? And if so, how early in the course do these observed grade relationships hold? Second, can we use partial social networks to help inform instructors and students in MOOCs? If the answer to these questions is true then it may be possible to develop effective social intervention systems that could use students' posting behaviors to flag students that need attention, or to generate strategic advice on where or how often to post questions. Section 2 provides some background on social network analysis in education. Section 3 describes the dataset we use in our work. In Sections 4 and 5 we present our analysis and results. And finally in section 6 we present our conclusions and discuss our future work.

2. BACKGROUND

2.1 MOOCs, Forums, Students Performance

According to Seaton et al. most of the time students spend on MOOCs is spent viewing the lecture videos, completing mastery assignments, and reading the discussion forums [21]. Very little time is spent on external or 'off-platform' activities. Thus, the discussion forums provide a rich and useful window into the students' primary course activities. Stahl et al. [24] illustrated how students collaborate to create knowledge through this interaction. They argued that students' forum activities are not only beneficial for the individual discussants but also serve to structure the class as a whole. Each student's activity level varies as does their impact on the course. Huang et al. for example, specifically investigated the behavior of high-volume posters in 44 MOOC-related forums. These 'super-posters' tend to enroll in more courses and generally perform better on average [12]. Moreover, by actively engaging in many conversations, they add to the overall volume of the course discussion and they tend to leave fewer questions unanswered in the forums. They also found that, despite their high output, these super-posters did not act to suppress the activity of other less-active users. Rienties et al. [19] examined the way in which students structure their social interactions online. They found that allowing students to self-select collaborators in a MOOC is more conducive to learning than random assignment of partners. In another study, Van Dijk et al[25] found that simple peer instruction is significantly less effective in the absence of a group discussion step, thus reinforcing the importance of a shared class forum.

Prior researchers have also examined the general dynamics of the student forums. Boroujeni et al. examined the relationship between students' temporal patterns, discussion content and social structures emerging from the forums [23]. They found that for MOOCs lasting eight weeks, the pace of students' posts remained high during the first 3 weeks and then tapered down gradually until the class ended. They also found that this pattern was affected by the assignment dates and other deadlines as well as the overall volume of the posts in each thread. Furthermore, they tracked the network attributes over time by using one-week network slices based upon a sliding window. The slice for each day of the course (d>6) was built from forum activities during the preceeding 7 days ([d-6, d]). For each network slice, the attributes included node counts, edge counts, average degree, density, etc. They found that, with the exception of density, the attributes decreased over time. Density, ratio of the number of edges in the graph and the number of edges possible, by contrast, increased sharply at the end of course. Zhu et al. explored a longitudinal approach to combine student engagement, performance, and social connections by applying exponential random graph models [29]. They analyzed the relationship between the social networks on a week-by-week basis and they found that students' individual assignment scores were all positively related to being more active in the social network.

Rosé et al.[20] examined students' evolving social interactions in MOOCs using a Mixed-Membership Stochastic Block model which seeks to detect partially overlapping communities. Their specific focus in the analysis was on identifying the students who were most likely to drop out. They found that it was possible to predict whether or not a student would drop out based upon their membership in a community. Students who actively participated in the forums early on in the course were less likely to drop out later on. Moreover, they found that one specific sub-community was much more prone to dropout than the remainder of the class. This suggests that the forum communities do align by stability and thus that social relationships can reflect the students' relative level of motivation as well as their overall experience in the course. This is akin to the 'emotional contagion' model used in the Facebook mood manipulation study by Kramer, Guillroy, and Hancock [16].

Dawson et al. [6] elaborated the use of social networks to provide guidance. They provided feedback to students and instructors based upon the students' *ego-social* network (i.e. their neighborhood). They explored differences in the network composition for low- and high-performing students to identify patterns of behaviours which may influence the students' learning. They found that the ego-social networks of low- and high-performing students had significant differences, and it was possible to identify different types of students based upon their ego-network. They also found that the instructors were equally likely to show up in high-performing students' local networks as in those of the low-performing students. Their results indicated that instructors could adjust their teaching methods based upon this network structure.

2.2 Communities

There has also been prior research specifically on how students connect within sub-communities and with the instructor. Insa et al. showed that in a traditional course (containing both face-to-face lectures and lab sessions), the student's seating position can affect their final grade [13]. They suggested that physical proximity to the instructor increased performance. According to Golder et al., an analysis of students' Facebook messages showed that the students will message one another more often during weekday afternoons than over the weekend [9]. This produced a distinct temporal pattern in their communication and community structure.

The motivation for any student to join a MOOC can vary widely. This can in turn create several distinct classes of participants with their own unique behaviors. Anderson et al., for example, argued that MOOC participants can be partitioned into 5 distinct categories based on the number of lectures that they watched and on the assignments that they submitted: viewers, solvers, all-rounders, collectors, and bystanders [1]. They also found that the more assignments a student completed and the more lectures that they viewed, the higher their final grade would be. Interestingly, while students who received a 'B' grade showed a small decrease in their homework submissions relative to 'A' students, the amount of time that those students spent watching lectures was substantially lower. In related work by Liu et al. however, the authors found that some of these behavioral differences were consistent with the students' cultural background which may affect not just their motivation but their expectations and habits [18].

Other authors have examined the relationship between students' academic performance and their social network relationships. Eckles et al. used network analysis on survey data to identify at-risk students who were more likely to drop out [7]. Kovanovic et al. analyzed how a student's relative centrality in their social network will affect their academic performance [15]. They found that more central students were typically higher performers than their less-connected peers. Finally, Zhang et al. constructed student social networks based upon the comments and replies that had been posted to the forum [28]. By analyzing the relative in- and out-degree of the vertices, they were able to identify a small amount of users who answered a large proportion of the questions. This allowed them to find key students in the course.

2.3 Student Behaviours

In their analysis of student behaviors, Anderson et al. found that the number of students who watched lecture videos and finished assignments decreased over the duration of the course, suggesting that some students changed their minds about the class or simply changed their habits during it [1]. Ye et al. performed a similar study, in which they examined a 10-week computer science MOOC [27]. At the end of week 4, 60% of the students who had only watched lectures but had not participated in other ways had dropped out of the course, while only 20% of the students who had submitted assignments and completed quizzes along with viewing had done so.

Given that a large number of MOOC registrants in a given course drop [1, 27], studying the causes of this dropout and preventing it is an important issue. Kloft et al. sought to predict dropout behaviors in a 12-week course based upon the students' click-stream data using a Support Vector Machine [14]. They identified two peak dropout points, one during the first two weeks of the course, and the second at the end of weeks 11 and 12. Students were unlikely to drop in the middle of the course and thus if they made it through the early stages and the final crunch then they would likely complete. Halawa et al. used a specialized definition of drop out as a student being absent from the course for more than 1 month or if they viewed less than half of the lecture videos [10]. With this definition they found that the percentage of students absent from the course sharply decreased from 36.4% to 13.8% after week 3. Hoskins , by contrast, focused exclusively on quizzes as performance-based indicators. They provided a web dashboard for students to self-assess their performance. By comparing students' self-assessments with their grades they found that low performing students tended to drop out more than their higher-performing peers [11].

Unlike the prior studies of students' performance on MOOCs we constructed a temporal social network structure to examine how and when MOOC students established their social connections with differently-performing peers, how their social connections changed over time, and the correlation between these community connections and their intermediate and final performance. We found MOOC students formed their social structures early in the course and that these relationships are stable over time.

3. DATA SET

In this study we used data from a 2013 course on "Big Data in Education" that was offered by the Teachers College at Columbia University and hosted on the Coursera platform. This was an 8-week course that was designed to cover all of the requisite material for a single-semester graduate-level course on Educational Data Mining (EDM) and Big Data analysis in education. This included studying core methods such as student modeling and introducing students to basic data collection and data analysis techniques such as logging and visualization. This iteration of the MOOC ran from October 24, 2013 to December 26, 2013. The course itself was structured around weekly lecture videos and individual assignments or quizzes which contributed to the students' final grade. The weekly assignments were structured around data analysis tasks with students being tasked with conducting some analysis discussed in class and then answering numeric or multiple-choice questions about it. The students were required to complete each assignment within two weeks of its being given out. They were also given up to three attempts per assignment.

The course had a total enrollment of over 48,000 students, but a much smaller number of active participants. 13,314 students watched at least one video while 1,242 watched all of them. A total of 1,380 students completed at least one assignment, and 778 made at least one post or comment in the forum. Of those students who made posts, 426 completed at least one class assignment A total of 638 students completed the online course and received a certificate (meaning that some were able to earn a certificate without participating in forums at all). In order to receive a certificate students were required to earn an overall grade average of 70% or above on the assignments [26].

4. METHODS

We began our analysis by clustering the count of students' submissions for each assignment by date in order to understand when students completed their assignments and how the submission patterns might indicate their working habits. Unsurprisingly the assignment submissions peaked right before each due date with few if any late submissions. To make our analysis consistent we broke the 8-week course into 2-week chunks and we split our analysis at weeks 2 (start), 4 (midterm), 6 (third quarter), and 8 (final). This decision was based upon the fact that students worked across weeks, and on prior literature that pegged the two- and four-week boundaries as crucial times for dropout (e.g. [14, 27]).

This partitioning yielded four distinct datasets representing the cumulative forum discussion up to that point in the class. We extracted a social network from each of these datasets using the same approach applied by Brown et al. [3, 2]. In this approach we generated a raw social network for the course where each node represents a single participant (student, TA, or instructor). We then labeled the student nodes with their cumulative performance up to the specified time step. Thus, the week 2 dataset was labelled using their cumulative performance up to the end of week 2. The Coursera forums operate as standard threaded forums. Users have the ability to start new threads by making an initial post. They can also add posts to the end of an existing thread or add a specific reply below a given post.

In order to build social network from the discussion forum, we treated participants as nodes and their communications as edges. More specifically, for each comment in a thread, we added a directed arc from the author's node to nodes representing the author of each comment that precedes it in the thread, with the exception of self-loops. So all of the contributors to a thread, including the originator, will be connected to one another. This approach is based upon the assumption that students read the thread *before* contributing to it and that a post represents a contribution to the whole conversation. The average length of each thread in our dataset was seven posts. Thus we treat each reply as evidence of an *implicit social connection* between the individual author and their conversational peers. Such implicit social relationships have been explored in the context of recommender systems to detect strong communities of researchers [4]. The resulting networks form a multigraph with each edge representing a single communicative act. As our goal is to focus on social relationships we then modified this graph by eliminating all isolated nodes, and by collapsing the parallel edges to produce a weighted undirected simple graph representing connections between students.

In addition to analyzing the connections between students, we also sought to analyze the impact of the instructional staff and the active hub students on their social structure. We therefore generated three different graphs for each of the datasets: *ALL* which is the complete graph with all non-isolated nodes; *Student*, which eliminates the instructional staff; and *No Hub*, which removes both the instructional staff and the highly active 'hub' students. Since MOOCs are an at-will course students often drop out and we cannot always distinguish intentional dropouts from unintentional failure. In one typical dataset, for example, more than 80% of the students received a grade of 0 [1]. Therefore we also constructed graphs for students with and without students who received a grade of 0. While it is true that the final grade is only accessible at the end of the course we do not believe that this limits the generality of our results. By identifying features that are consistent with 0 performance we can develop predictive models that will work in real-time.

4.1 Best-Friend Regression

Fire et al. modeled students' social interactions for grade prediction in a traditional classroom [8]. They found that in traditional classes the students' grades are closely correlated with those of their closest neighbor or "best friend". That research was based upon self-reported relationship data, but Brown et al. were able to show that it also applied in an online context [2]. In that analysis they used the weighted network to identify each students' "Best Friend" (BF) or closest peer by connections. They then showed that the same result held for this network structure as well.

4.2 Community Detection

We applied the Girvan-Newman algorithm to find social clusters within our graph. In order to identify the ideal number of clusters we used the "natural cluster number" approach described in [3]. That approach is based upon the modularity score of candidate clusters. Given a graph that has been clustered into sub-communities, the modularity of the graph is measured by the ratio of intra-cluster to inter-cluster connections, that is, how strongly individual students are associated with their cluster associates relative to the rest of the class. Graphs with high modularity have very strong within-cluster connections and relatively sparse connections across the groups. As the graphs are partitioned into smaller and smaller communities the modularity score will grow rapidly until we reach an inflection point or a point of diminishing returns at which point each additional sub-cluster makes little difference to or even reduces the modularity score. In the natural cluster approach, we iteratively cluster the graph into higher numbers of communities and plot the modularity score over number of clusters. We then examine this curve to find the inflection point and use that value. This is an exploratory approach similar to exploratory Principal Components Analysis.

4.3 MOOCs, Forums, Student Performance

In MOOCs, the class forum is typically the only official way for students to communicate with the instructors and with each other. Thus, their activities on the forum represent a mostly-complete record of their communicative actions and it represents the best record of their questions and interests. So the dynamic of student forum activities represent their real-time learning status. In order to investigate the dynamics of the students' forum activities and their relationship with the students' social networks, we extracted the number of posts and comments, the number of forum users (who wrote posts) and the number of threads added on a biweekly basis. We then analyzed the numbers in each two-week pair to find the scale of the social network in each case. We also explored how the social aspects of the discussion forum changed over time, by calculating density, degree, average path, diameter and other basic metrics. These network attributes represent the evolving network structure. Furthermore, we compared the scale of the dynamic networks and the network structures to determine when the social networks stabilized. Finally, we analyzed the average number of changes in the neighbors for each student to learn how students selected their communities biweekly.

5. RESULTS & DISCUSSION

Table 1 shows the order (number of nodes) and size (number of edges) of the graphs that we obtained at each cutoff point. While the graphs grew monotonically in order and size over the duration of the course, most of the connections between the students were already established by the end of week two. That is, the basic network structure, if not its weight, was set early on.

Table 1: Graph order and size for each cutoff.

	Nodes	Edges	Comments
Week2	645	14,050	2,472
Week4	693	15,142	3,231
Week6	725	16,346	3,833
Week8	754	17,004	4,260

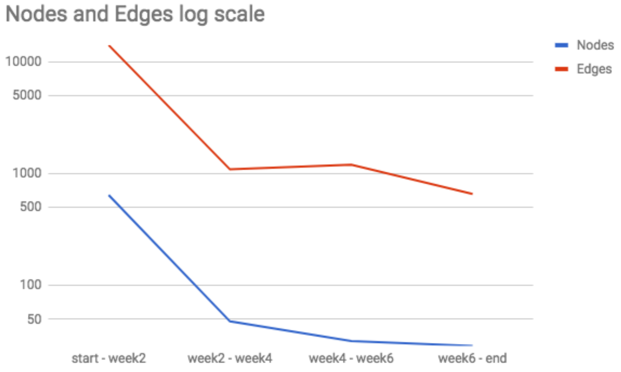


Figure 1: New participating students and connections every two weeks

At the end of the course, there were 55,179 registered users, yet the final course graph contained only 754 participants, 751 of whom were students with 1 instructor and 2 Teaching Assistants. Additionally, 304 of the 751 students obtained a zero grade at the end of the course while 447 received non-zero grades. Some of the forum participants did not complete any assignments but still chose to discuss the course topics with others. By the same token, some of the students who completed work in this course did not participate in the forum at all. There were 1,381 students who received a non-zero final grade; 934 of these did not post in the forum, while 304 zero final grade students did. It is conceivable that when the students met with problems, they chose to ask questions online, but participation in the course forum was not a necessary condition for completion.

Figure 1 shows the number of new participants and new connections added into the social network every two weeks. We applied log scale for the y-axis to make the chart more readable. As these results illustrate almost all students and instructors had established their connections in this course by the end of week two and only a few new connections were made after that time. Additionally, the total number of posts/comments made was 4,260; 2,472 of them (or 58%) had been made at the end of week 2. In our later analysis, we defined a distinct type of 'social connection' post, which includes student-initiated introductions to the class as well as attempts to set up general social connections via Facebook groups, Linkedin links, or other mechanisms. As a results, we collected 182 'social connection' type of student posts. However, even if we discount those 'introduce yourself' comments, it still shows that most of the posting activity happened at the beginning of the course. One potential explanation for this is that the students, particularly those who did not plan to obtain a certificate, did most of their work early and subsequently lost interest. Or, some of the students worked in spurts and did not fit the schedule over time. An ongoing analysis of the forum content has shown that a number of the posts are also about early issues such as course logistics and software, problems which may be less relevant later on. Irrespective of the cause, the social structure is well established early enough that information based upon it can be used to advise students before it is too late.

5.1 Best-Friend Regression

As part of our analysis we also replicated the Best-Friend comparison used by Brown et al. Here we identified each student's closest neighbor in the course, ignoring teaching staff, and we calculated a direct correlation between their grades and those of their best friends. Because the data was non-normal we used Spearman's Rank Correlation Coefficient (ρ), a non-parametric measure of association [22, 5]. Our results are shown in Table 2. Because week 8 is the last week of the course, the intermediate grade is the final grade.

Table 2: Correlation and p-values for Best Friends analysis.

	intermediate grade		final grade	
	ρ	p	ρ	p
Week2	0.25	< 0.001	0.27	< 0.001
Week2_non0	0.086	0.12	0.093	0.08
Week4	0.313	< 0.001	0.339	< 0.001
Week4_non0	0.145	0.005	0.158	0.002
Week6	0.42	< 0.001	0.437	< 0.001
Week6_non0	0.25	< 0.001	0.295	< 0.001
Week8	NA	NA	0.44	< 0.001
Week8_non0	NA	NA	0.29	< 0.001

As shown in Table 2, the students' grade and their best friends' grades, both final and intermediate grades for each bi-week, were strongly correlated, ρ was high, and significant $p < 0.001$. However, the correlation ws affected by the clusters of 0 grade students. After removing these students, the correlations did not hold at a statistically-significant level until the middle of the course. After week four, we found a moderate correlation, $\rho = 0.295$, $\rho = 0.25$, $\rho = 0.29$ and $p < 0.001$. Thus, the relationship between students' grades and those of their best friends were consistent from the traditional face-to-face class to MOOC but not immediately. Our results show that MOOC students, except those who did not submit any assignments, performed similarly to their closest peers.

5.2 Community Detect

Figure 2 provides an example of the modularity curves both with and without zero-score students. We selected natural cluster numbers by finding the inflection points for modularity score. Table 3 shows the selected number of natural clusters based on each week's intermediate grade and table 4 shows the number of clusters based upon the final grade. From table 3 and table4, we found the maximum modularity score for clusters decreases over time. As the modularity score is designed to measure the cleanliness of dividing the network into clusters, these results indicate that the connections between the individual students become more sparse

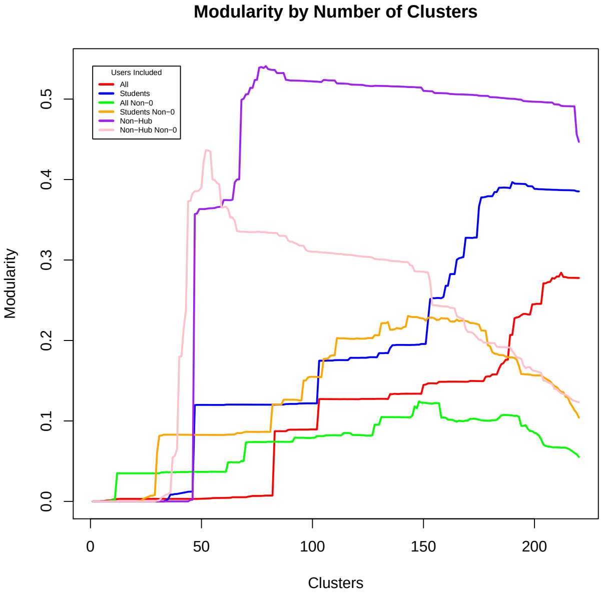


Figure 2: Modularity by Number of Clusters for Week 8

over time while the connections between the clusters of students become more dense as the course progresses.

Interestingly, the curves for the ALL and Hub Student graph are extremely similar, which indicates that hub-students were those who kept a close connection with instructor and TAs. As we anticipated, the non-zero students are the largest group of students. The social network graph shows that many of the zero-score students were only connected with other zero-score students which supports our argument that poor performing students are likely to connect with others at the same performance level.

Table 3: Modularity and number of clusters for each graph with intermediate grade

Graph Type	Week2	Week4	Week6
All Modularity	112 0.346	177 0.327	200 0.276
All_non0 Modularity	56 0.276	100 0.195	121 0.122
Students Modularity	119 0.414	129 0.419	172 0.393
Students_non0 Modularity	63 0.436	97 0.346	125 0.266
Nonhub Modularity	63 0.590	67 0.590	69 0.553
Nonhub_non0 Modularity	43 0.613	41 0.490	55 0.396

Table 4: Modularity and number of clusters for each graph with final grade

Graph Type	Week2	Week4	Week6	Week8
All	112	177	200	212
Modularity	0.346	0.327	0.276	0.284
All_non0	56	135	149	173
Modularity	0.257	0.202	0.161	0.103
Students	119	129	172	184
Modularity	0.414	0.419	0.393	0.390
Students_non0	63	109	130	169
Modularity	0.439	0.351	0.304	0.224
Nonhub	63	67	69	79
Modularity	0.590	0.580	0.553	0.541
Nonhub_non0	43	45	49	52
Modularity	0.570	0.478	0.407	0.437

Table 5: Average Dynamic Cluster Changes with final grades

finalgrade	all			all_non0		
week	lost∗	gain∗	overlap∗	lost	gain	overlap
2-4	11.7	1.75	29.6	8.46	2.63	9.94
4-6	1.75	1.75	28	2.53	17.9	9.3
6-8	1.9	3.27	26.74	9.86	36.3	16.9
	students			students_non0		
week	lost	gain	overlap	lost	gain	overlap
2-4	2.05	9.47	30.7	19.3	2.61	11.4
4-6	9.7	1.55	28.77	3.8	2.96	9.42
6-8	1.64	2.72	27.7	2.72	8.94	9.34

lost: average number of lost neighbors
gain: average number of new neighbors
overlap: average number of the same neighbors

To assess cluster stability, we also calculated student-centric cluster similarity metrics for the graphs. Tables 5 and 6 show the average number of neighbors that each student loses, gains, or retains in their cluster from week to week. That is, it shows how many former friends are now in a different group, how many new friends are added, and how many stay the same. These figures are shown for weeks 2-4, 4-6, and 6-8 for the all but the no-hub graphs. We excluded the no-hub graphs from our analysis because the models were constructed week by week, the specific hub students did change over time(22% hub group changed from week2 to week8). We also generated the metrics for the social networks based upon the final grades and the weekly cumulative grades. As the tables illustrate, the clusters lost members in each week with the losses being highest in the jump from week 2 to week 4, when the network is still growing quickly. In the later weeks, the losses were smaller, particularly in weeks 4-6. And, for all but the All_NonZero graph, the students gained few new neighbors, with most of the neighbors being retained. As discussed above, the number of clusters increased as the course went on. As these tables indicate the new clusters were generally subsets of the prior clusters and did not present a remix of the prior neighborhoods. The lone exception was the All_NonZero graph which had substantial gains in weeks 4-6 and 6-8. This suggests that the lurkers and other non-certification-seeking students are an important factor in the stability of the social networks; thus, discarding them has a notable effect. However, more analysis is required to understand just how they engender this stability and just how widely distributed they are in the clusters.

Table 6: Average Dynamic Cluster Changes with Intermediate Grades

intergrade	all			all_non0		
week	lost	gain	overlap	lost	gain	overlap
2-4	11.7	1.75	29.6	14.6	20.2	17.1
4-6	1.75	1.75	28	6.87	50	28.8
6-8	1.9	3.27	26.74	41.7	15.3	37.7
	students			students_non0		
week	lost	gain	overlap	lost	gain	overlap
2-4	2.05	9.47	30.7	20.5	3.2	11.2
4-6	9.7	1.55	28.77	3.28	17.5	10.3
6-8	1.64	2.72	27.7	17.3	8.2	10

5.3 Student Performance & Motivation

According to the social network graph, students clustered into different clusters based on their connections and their performance. In order to examine the grade distribution of each cluster, we applied the Kruskal-Walls(KW) test to evaluate the correlation between clusters and performance. The KW test is a non-parametric rank-based similar to the common Analysis of Variance [17]. The result for each graph shown in table 7 - 8 while the 'F' column value is Chi_square. We can see that for nonzero score students, their performance was highly related with their clustered friends, but when all students are included, the relationship becomes weak. This result supports our hypothesis that students will connect with similar performers, instead of helping poor performing students or learning from good ones [26].

Table 7: KW test with intermediate grade

Graph Type	Week2		Week4		Week6	
	F	P	F	P	F	P
all	207	< 0.001	270	< 0.001	315	< 0.001
all_non0	74	0.04	133	0.07	129	0.25
students	218	< 0.001	228	< 0.001	285	< 0.001
students_non0	55	0.69	118	0.06	142	0.12
nonhub	134	< 0.001	171	< 0.001	182	< 0.001
nonhub_non0	53	0.1	47	0.18	90	0.001

Table 8: KW test with final grade

Graph Type	Week2		Week4		Week6	
	F	P	F	P	F	P
all	210	< 0.001	273	< 0.001	319	< 0.001
all_non0	70	0.19	154	0.1	168	0.12
students	223	< 0.001	239	< 0.001	293	< 0.001
students_non0	80	0.06	127	0.1	164	0.01
nonhub	145	< 0.001	179	< 0.001	190	< 0.001
nonhub_non0	44	0.2	58	0.06	67	0.03

Table 9: Forum attributes over time

Attribute	Week2	Week4	Week6	Week8
Posts	2514	3231	3833	4233
Users	659	707	742	770
Threads	345	460	545	597

Table 9 is representative of the evolution of the forum attributes over the 2 week intervals. The overall number of posts, threads, and users increase over time. From the table, we can see that the increase in the number of posts and threads is stable from course start to end. By the end of week 2, 59.4% of the posts had been added to the data and 57.8% of the threads were started in the course forum. However, considering the number of users, 85.6% of the total forum users showed up by week 2. So, by one quarter of the way through the class, most of the users had already showed up in the forum, but fewer than 60% of posts and thread had been initiated. Table 10 shows that the values of the network attributes don't have clear changes which may indicate that the root social network structure doesn't change after week 2. Thus, the dynamics of the forum attributes are consistent with our findings for the best friends and community analysis over time, that the student forum social network structure will develop as soon as week 2 and will then become stable, with the small communities and best friends only getting stronger.

Table 10: Network attributes over time

Attribute	Week2	Week4	Week6	Week8
Degree	21.783	21.850	22.546	22.552
Density	0.034	0.032	0.031	0.030
Avg_path	2.607	2.535	2.492	2.490
Diameter	7	7	7	7
Connected component	82	88	89	98

6. CONCLUSION

Our goal in this paper was to address the potential utility of social network information to guide students and instructors in MOOCs. As prior work has shown, students' final social network structures, particularly their closest neighbors or "best friends" and their sub-clusters, can be analyzed to predict their performance. However, in order to provide meaningful guidance, or to help students and instructors improve their performance before it is too late, it is necessary to show that we can extract useful information from partially-formed social networks. In this paper we have shown that the structure of the students' social networks can be analyzed to predict their performance even by the second week in the course.

Consistent with the prior literature, we found that students are most closely associated with similarly-performing peers and it is possible to predict students' performance based upon their closest neighbors in the graph. Therefore, good students are not necessarily connecting closely with poorer performers, or spreading their help evenly across the class. These results hold even if we remove the instructional staff, hub students, and zero-grade students from the course.

These results suggest that it could be possible to use forum data to identify isolated students or poorly-performing subcommunities that are in need of help. It might also help provide guidance to students who may not be seeking help from the right places. By identifying students who are not isolated, but who are not necessarily getting help from good peers, we may be able to intervene to not only improve their individual standing but also to improve the (social network) structure of the course as a whole. These results also suggest that we should consider mechanisms to encourage more distributed feedback, such as explicit rewards for peer tutoring.

Interestingly, we found that students' social behaviours are consistent because, while students continue to contribute to the course over time, the social structure of the course is established relatively early. More than half of the forum posts are made in the first two weeks of class. And few students begin to participate on the forum after that point. It is not the case that we have a dynamic graph which can be analyzed differently at each stage. Rather, it appears that the basic structure of the social relationships are fixed early and then only grow stronger over time. While more analysis is required to determine why this occurs, it suggests that students' initial impressions or choices have a strong impact on their performance and that interventions which are designed to change those habits may be beneficial. One avenue of research that we are currently pursuing is to analyze the content of the individual posts. If we can detect a change in the nature or structure of the content or of the topics being considered it might help to explain why the students' progress appears to taper off so dramatically. At the same time we plan to experiment with evaluating metrics of this type for blended courses to see if similar dynamic results hold in blended face-to-face and online contexts.

Furthermore, our results indicate that a social network analysis of the discussion forum data brings an unprecedented opportunity for instructors to visualize students' social structures and to form learning networks which allow them to make changes to their teaching plans over time. For nonzero grade students, the correlation between students' grades and their best friends' grades is not reliable during the first 4 weeks of the course. However network features may be useful for early detection of at-risk students. Real-time ego-networks may also explain how low performance is related to connections to other low performing students. This suggests that it may be useful to incentivize high performing students to make connections with lower performing student threads.

7. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1418269: "Modeling Social Interaction & Performance in STEM Learning" Yoav Bergner, Ryan Baker, Danielle S. McNamara, & Tiffany Barnes Co-PIs.

8. REFERENCES

[1] 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*, pages 687–698. ACM, 2014.

[2] R. Brown, C. Lynch, Y. Wang, M. Eagle, J. Albert, T. Barnes, R. Baker, Y. Bergner, and D. S. McNamara. Communities of performance & communities of preference. In *EDM (Workshops)*, 2015.

[3] R. Brown, C. F. Lynch, M. Eagle, J. Albert, T. Barnes, R. Baker, Y. Bergner, and D. S. McNamara. Good communities and bad communities: Does membership affect performance? In *EDM (Workshops)*, 2015.

[4] E. Choo, T. Yu, M. Chi, and Y. Sun. Revealing and incorporating implicit communities to improve recommender systems. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 489–506. ACM, 2014.

[5] P. Dalgaard. *Introductory Statistics with R*. Springer Verlag New York Inc., 2002.

[6] S. Dawson. âĂŸseeingâĂŹthe 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.

[7] J. E. Eckles and E. G. Stradley. A social network analysis of student retention using archival data. *Social Psychology of Education*, 15(2):165–180, 2012.

[8] M. Fire, G. Katz, Y. Elovici, B. Shapira, and L. Rokach. Predicting student examâĂŹs scores by analyzing social network dataanderson. In *International Conference on Active Media Technology*, pages 584–595. Springer, 2012.

[9] S. A. Golder, D. M. Wilkinson, and B. A. Huberman. Rhythms of social interaction: Messaging within a massive online network. *Communities and technologies 2007*, pages 41–66, 2007.

[10] S. Halawa, D. Greene, and J. Mitchell. Dropout prediction in moocs using learner activity features. *Experiences and best practices in and around MOOCs*, 7, 2014.

[11] S. L. Hoskins and J. C. Van Hooff. Motivation and ability: which students use online learning and what influence does it have on their achievement? *British journal of educational technology*, 36(2):177–192, 2005.

[12] 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*, pages 117–126. ACM, 2014.

[13] D. Insa, J. Silva, and S. Tamarit. Where you sit matters how classroom seating might affect marks. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, pages 212–217. ACM, 2016.

[14] 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.

[15] V. Kovanovic, S. Joksimovic, D. Gasevic, and M. Hatala. What is the source of social capital? the association between social network position and social presence in communities of inquiry. 2014.

[16] A. D. Kramer, J. E. Guillory, and J. T. Hancock. Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences*, 111(24):8788–8790, 2014.

[17] 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.

[18] Z. Liu*, R. Brown*, **C. F. Lynch**, T. Barnes, R. Baker, Y. Bergner, and D. Mcnamara. Mooc learner behaviors by country and culture; an exploratory analysis. In T. Barnes, M. Chi, and M. Feng, editors, *Proceedings of the 2016 Conference on Educational Data Mining.* International Educational Data Mining Society, 2016.

[19] B. Rienties, P. Alcott, and D. Jindal-Snape. To let

students self-select or not: that is the question for teachers of culturally diverse groups. *Journal of Studies in International Education*, 18(1):64–83, 2014.

[20] 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 *Proceedings of the first ACM conference on Learning@ scale conference*, pages 197–198. ACM, 2014.

[21] D. T. Seaton, Y. Bergner, I. Chuang, P. Mitros, and D. E. Pritchard. Who does what in a massive open online course? *Commun. ACM*, 57(4):58–65, Apr. 2014.

[22] P. Sedgwick. Spearman's rank correlation coefficient. *BMJ: British Medical Journal (Online)*, 349, 2014.

[23] M. Shirvani Boroujeni, T. Hecking, H. U. Hoppe, and P. Dillenbourg. Dynamics of mooc discussion forums. In *7th International Learning Analytics and Knowledge Conference (LAK17)*, number EPFL-CONF-223718, 2017.

[24] G. Stahl, T. Anderson, and D. Suthers. Computersupported collaborative learning: An historical perspective, 2006. *Cambridge handbook of the learning sciences*, pages 409–426, 2006.

[25] L. Van Dijk, G. Van Der Berg, and H. Van Keulen. Interactive lectures in engineering education. *European Journal of Engineering Education*, 26(1):15–28, 2001.

[26] Y. Wang and R. Baker. Content or platform: Why do students complete moocs? *Journal of Online Learning and Teaching*, 11(1):17, 2015.

[27] C. Ye and G. Biswas. Early prediction of student dropout and performance in moocss using higher granularity temporal information. *Journal of Learning Analytics*, 1(3):169–172, 2014.

[28] J. Zhang, M. S. Ackerman, and L. Adamic. Expertise networks in online communities: structure and algorithms. In *Proceedings of the 16th international conference on World Wide Web*, pages 221–230. ACM, 2007.

[29] M. Zhu, Y. Bergner, Y. Zhang, R. Baker, Y. Wang, and L. Paquette. Longitudinal engagement, performance, and social connectivity: a mooc case study using exponential random graph models. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 223–230. ACM, 2016.

QuanTyler : Apportioning Credit for Student Forum Participation

Ankita Bihani
Stanford University
Stanford, USA
ankitab@stanford.edu

Andreas Paepcke
Stanford University
Stanford, USA
paepcke@cs.stanford.edu

ABSTRACT

We develop a random forest classifier that helps assign academic credit for a student's class forum participation. The classification target are the four classes created by student rank quartiles. Course content experts provided ground truth by ranking a limited number of post pairs. We expand this labeled set via data augmentation. We compute the relative importance of the predictors, and compare performance in matching the human expert rankings. We reach an accuracy of 0.96 for this task. To test generality and scalability, we trained the classifier on the archive of the Economics Stack Exchange reputation data. We used this classifier to predict the quartile assignments by human judges of forum posts from a university Artificial Intelligence course. Our first attempt at transfer learning reaches an average AUC of 0.66 on the augmented test set.

Keywords

Online Discussion Forum, MOOCs, residential courses, random forest, credit computation, online learning, transfer learning, instructor support, collaborative learning, grading, crowdsourcing, forum assessment.

1. INTRODUCTION

Massively Open Online Courses (MOOCs) have in past years provided content to populations outside traditional venues of higher education. For these settings, online forum facilities that are built into the course delivery platforms, such as Coursera and Open edX are the primary means of communication among learning peers, and for interacting with instructors.

Beyond the practical needs for coordinating logistics in geographically distributed settings, online discussion forums can serve pedagogical goals as well. Online asynchronous discussion forums provide the basis for collaborative learning, which enhances critical thinking [10]. Students answering each others' questions can be helpful for all parties [14].

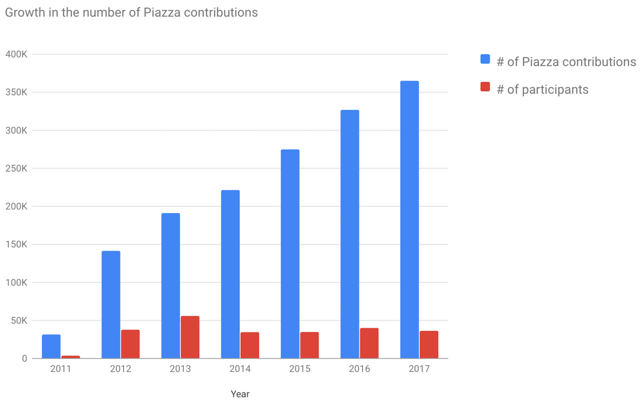


Figure 1: Number of Piazza forum contributions and participants per year for courses at our University

This support function is particularly useful in Science and Engineering courses. But as discussion centric humanities courses embrace distance learning, discussions on online forums will likely gain even more prominence.

However, it is not only in the context of distance learning that forum facilities have found uses. Even when in person class time is available, many residential college courses have adopted the tool. The need for students to ask questions, voice concerns, or to point out errors in course material are as salient in residential settings as they are in less traditional situations, such as distance learning [4]. Figure 1 shows the rapid growth in the volume of contributions per year to Piazza, just one of the several available online forum tools in a large private university. Despite the fact that the total number of Piazza participants were roughly the same from 2012 to 2017 (with a slight peak in 2013), the total volume of contributions increased monotonically. It is possible that this rapid increase in the volume of contributions per year on Piazza stems from the increasing popularity and growing adoption of the Piazza forum among students and instructors for collaborative discussions.

Given the importance of collaborative discussion in the learning process at both the theoretical and empirical level, instructors in at least some universities are assigning between 1% and 25% of their course grading component to online forum contribution. Two primary challenges arise when apportioning course credit to reward students' forum contri-

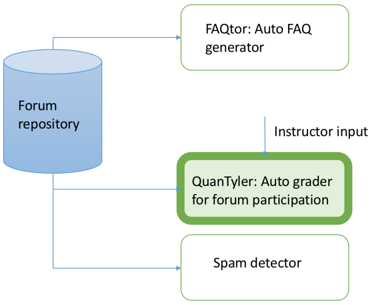

FAQtor: Auto FAQ generator
Forum repository
Instructor input
QuanTyler: Auto grader for forum participation
Spam detector

Figure 2: Block diagram of our proposed framework around forum facilities.

bution. First, students can attempt to game the system. On surveying some instructors, we learnt about instances of students copying a peer's forum posts, adding spaces or other innocuous characters to fool automated contribution counters. Thus, the system needs to be able to flag such instances.

A second, more complicated problem is that of apportioning fair credit to the students at the end of the quarter. Forum contributions take many forms. Asking an insightful or intriguing question can contribute as much to the course as providing answers. Taking the time to view other students' contributions is a contribution as well. However, for courses with hundreds of students, manual assessment of every forum post by each student in order to assign a forum participation score is not feasible. On surveying some instructors, we learnt that they instead develop ad–hoc formulae over the participation statistics provided by the forums, hoping to capture the right signals. This practice can not only lead to non-uniform grading (based on diverging intuitions) across courses, but also fail in rewarding students with a fair forum participation credit commensurate with their effort.

In addition to the above two challenges, there is untapped potential from today's use of forum facilities. As courses are offered repeatedly over the years, a treasure of course knowledge accumulates in forum archives. The detection of high value forum contributions can inform content selection from such archives.

In an effort to address these problems we are developing a coherent system for boosting the value of online discussion forums. Figure 2 shows a block diagram of our proposed system. In this paper, we focus on *QuanTyler*, the module responsible for helping with automatic forum credit apportioning. This component is highlighted in the figure. We plan to make the operation of *QuanTyler* customizable by instructors. For instance, instructors will be able to decide the granularity of partitioning the class into their quantiles of choice.

We begin with describing how we used human judgments to establish ground truth of what a 'good' and credit-worthy forum contribution looks like. This ground truth is used for measuring success, and for training the models. At the heart of our contribution are three experiments whose outcomes are required to inform the development of the *QuanTyler* module. These experiments are outlined below.

In the first experiment, we explore how students can be classified into quantiles based on their forum contributions, such that the implied ranking matches the ground truth. We show the hyperparameters needed to make a Random Forest classifier work well in support of the post evaluation task. We reached a high AUC measure in this task.

However, obtaining human judgments is expensive. At the same time, this requirement for human judgment would limit the ability to create classifiers for many courses. To break out of this confinement, we examine how a much larger source of labels for a forum-like enterprise might be used for training, and to test generalizability.

To this purpose we used *Stack Exchange*, [2] which is an online Q & A platform with millions of users. Stack Exchange is partitioned into sites for varying disciplines. We chose the Economics archive [1], and used it as a source for attempting transfer learning. In our second experiment we trained a random forest model on Stack Exchange reputation data, and tried predicting the quality ratings of human expert-rated forum posts in an Artificial Intelligence (AI) class. While not as good a classifier as the one trained on the forum data itself, this first attempt at transfer learning reached an $AUC = 0.66$, which we hope to improve further going forward. However, the data from Stack Exchange cannot be used in its raw form to build a classifier, and we will cover the required processing in Section 8.

In our third experiment, we demonstrated that (at least one of) the ad–hoc formulas currently deployed at our university diverges significantly from human experts' judgment.

2. RELATED WORK

Online discussion forums empower students and instructors to engage one another in ways that promote critical thinking, collaborative problem solving, and knowledge construction [20, 17]. Research has shown that linking some form of assessment to forum participation is an important element in promoting and enhancing online interactivity [16, 28].

Quantitative methods for content analysis are most widely used in assessing effective forum participation. [7] presents an overview of 15 different content analysis instruments used in computer supported collaborative learning (CSCL) studies.

The model proposed by [12] is a common starting point in many CSCL studies. In [12], the author presented a framework and analytical model to evaluate computer-mediated communication (CMC). The analytical model was developed to highlight five key dimensions of the learning process exteriorized in messages: *participation, interaction, social, cognitive and metacognitive dimensions.* Although this model provides an initial framework for coding CMC discussions, it lacks detailed criteria for systematic and robust classification of electronic discourse [13].

Many researchers have strongly endorsed Social Network

Analysis as a key technique in assessing the effectiveness of forum interactions [6, 29, 8]. Social Network Analysis is a research methodology that seeks to identify underlying patterns of social relations based on the way actors are connected with each other [25, 22].

In [18], the authors discuss a conceptual framework for assessing quality in online discussion forums. Drawing on previous work [12, 19, 9], the authors propose three broad categories of criteria for assessing forum participation: *content*, which demonstrates the type of skill shown by the learners, *interaction quality*, which looks at the way learners interact with each other in a constructive manner, and *objective measures*, which highlight the frequency or participation. These three broad criteria are further divided, resulting in a total of 11 criteria. In order to support educators, the framework outlines a further sub classification, clearly indicating what may be a poor, satisfactory, good or excellent performance against each criterion. The primary limitation of this study is that manual assessment by instructors is not feasible in courses with hundreds of students.

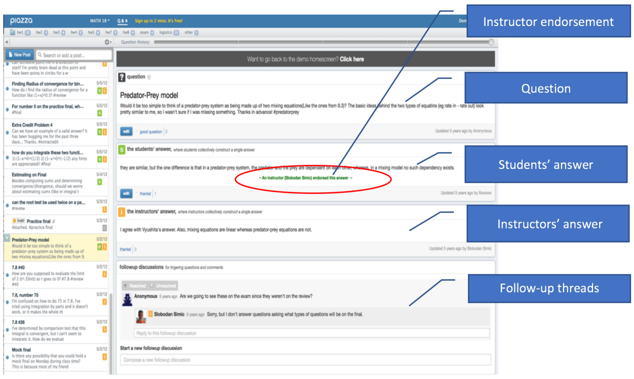
In [24], the authors adopt a content analysis approach and develop a coding scheme to analyze students' discussion behaviors in order to categorize them as active, constructive or interactive. However, the authors do not discuss how to apportion forum participation credit based on the behaviors depicted. One of their findings shows that higher quantity of participation in the MOOC discussion forums is associated with higher learning gains. In coherence with this finding, we also include participation count as one of our potential predictors.

To the best of our knowledge, the most closely related work to our paper are [21] and [23].

In [21], the authors present the use of Social Network Analysis (SNA) to examine the structure and composition of ties in a given network, and provide insights into its structural characteristics. In particular, the authors rely on two types of networks: interaction network between students in a course, and the network of terms used in their interactions. The dynamic visualization of interaction between participants and the groups or communities formed can help the instructors rank students based on their centrality in the students' interaction network. Visualizing the network of terms used in an online discussion forum can be used to compare the interest of different students and their relative engagement.

In [23], the author proposes the use of the following metrics to assess forum participation: initiative, effectiveness–depth, effectiveness–breadth, value, timeliness, participation, scholarship, style, and instructor points. Our system explicitly or implicitly covers most of these measures and augments them further by adding the crucial element of social network analysis to assess forum participation.

In contrast with both the aforementioned contributions, each of which focuses on specific aspects for assessing forum participation, our approach for assessing a student's contribution uses a combination of quality measures, quantitative measures, engagement level measures and also measures from social network analysis. The intent is to provide a holistic view of each student's contribution. We develop a system that the instructors can customize and easily use for apportioning forum participation credit.



start? I'm pretty brain dead at this point and have been going in circles for a w

Finding Radius of convergence for bin... 5/5/12
How do I find the radius of convergence for a function like (1+x)^0.5? #review

For number 5 on the practice final, wh... 5/5/12
#final

Extra Credit Problem 4 5/5/12
Can we have an example of a valid answer? It has been bugging me for the past three days... Thanks. #extracredit

how do you integrate these two functi... 5/5/12
1) (1-x^4)^(1/2) 2) (1-x^4)^(-1/2) any hints are appreciated!! #final

Estimating on Final 5/4/12
Besides computing sums and determining convergence/divergence, should we worry about estimating sums (like in integral t

can the root test be used twice on a pa... 5/3/12
#review

Instr Practice final 5/3/12
Attached. #practice.final

Predator-Prey model 5/2/12
Would it be too simple to think of a predator-prey system as being made up of two mixing equations(Like the ones from 9.

7.8 #40 5/2/12
How are you supposed to evaluate the limit of 2 (t^.5)ln(t) as t goes to 0? #7.8 #review #40

7.8, number 75 5/2/12
I'm confused on how to do 75 in 7.8. I've tried using integration by parts and it doesn't work, or it makes the whole th

7.8 #26 5/2/12
I've determined by comparison test that this integral is convergent, but I can't seem to integrate it. How do we evaluat

Mock final 5/2/12
Is there any possibility that you could hold a mock final on Monday during class time? This is because most of my friend

Want to go back to the demo homescreen? **Click here**

question

Predator-Prey model

Would it be too simple to think of a predator-prey system as being made up of two mixing equations(Like the ones from 9.3)? The basic ideas behind the two types of equatins (eg rate in - rate out) look pretty similar to me, so I wasn't sure if I was missing something. Thanks in advance! #predatorprey

Updated 5 years ago by Anonymous

the students' answer, *where students collectively construct a single answer*

they are similar, but the one difference is that in a predator-prey system, the predator and the prey are dependent on eachother, whereas, in a mixing model no such dependency exists.

~ An instructor (Slobodan Simic) endorsed this answer ~

Updated 5 years ago by Xxxxxxx

the instructors' answer, *where instructors collectively construct a single answer*

I agree with Vyushita's answer. Also, mixing equations are linear whereas predator-prey equations are not.

Updated 5 years ago by Slobodan Simic

followup discussions *for lingering questions and comments*

Anonymous 5 years ago Are we going to see these on the exam since they weren't on the review?

Slobodan Simic 5 years ago Sorry, but I don't answer questions asking what types of questions will be on the final.

Reply to this followup discussion

Start a new followup discussion

Compose a new followup discussion

Figure 3: Sample annotated screenshot of the Piazza forum facility.

3. CURRENT PRACTICE

Many universities use the Piazza forum facility [27] for asynchronous online discussions. In order to provide context for the experiments below, we provide a brief overview of this tool.

Piazza is a Q&A web service for online discussions, where users can ask questions, answer questions, and post notes. The user interface contains a dynamic list of posts, which are question titles followed by a snippet of lines from the post. For every question, there is a placeholder for the instructor's answer, which can only be edited by instructors. There is also a students' answer section where students *collaborate* to construct a single answer. Students can *upvote* each others' questions or answers. Instructors can also *endorse* good questions and answers, which are then highlighted as instructor endorsed. There is also a discussion segment for follow-up threads. Figure 3 shows a snapshot of the Piazza discussion forum.

On surveying several instructors who consider Piazza forum participation in their grading scheme, we found that most rely on the basic quantitative statistics that the forum machinery readily offers. The following were some of the grading schemes that are currently used by instructors at our university for awarding forum participation grades:

Scheme 1: In this scheme, scores of each student were calculated using the following formula:

$$\begin{aligned}\texttt{Score} = 1 * (\texttt{no\_questions\_asked}) + \\ 4 * (\texttt{no\_questions\_answered}) + \\ 0.5 * (\texttt{other\_contributions})\end{aligned} \tag{1}$$

Scheme 2: In this scheme, scores of each student were calculated using the following formula:

$$\begin{aligned}\texttt{Score} = 3 * (\texttt{no\_questions\_answered}) + \\ 1 * (\texttt{no\_followups})\end{aligned} \tag{2}$$

Anyone above the 90th percentile received full credit, and all the other students received a score of 0.
Scheme 3: Award full credit if at least one forum contribution was made, and the student was online on the forum for at least x number of days, or viewed at least y posts. Here x and y were set by the instructor using intuition.
Scheme 4: Award full credit to a student if they made at least one contribution to the forum.

All the above methods rely solely on the basic statistics directly provided by Piazza [27]. The concern, however, is whether these methods accurately and meaningfully award credit to the deserving students. Following are two major limitations of using the current grading schemes:

- *Lack of quality measures :* All the 4 grading schemes described above overlook the quality of contributions. This exclusion negatively impacts the grades of the students who post few, but very high quality contributions. More importantly, relying solely on the quantity of the contributions encourages posts that do not constitute meaningful forum participation. This behavior, in turn, can cause the forum's quality to devolve.
- *Reward not proportionate to effort :* Most of these schemes fail to award credit proportionate to the amount of effort and time the student invested. For instance, using the third or fourth scheme means that two students with vastly varying quantity and quality of contributions would be awarded the same score. Concretely, let us consider two students A and B. Student A made only one forum contribution during the course by posting a "+1" to another student's question. However, student B regularly made meaningful forum contributions throughout the quarter. Using grading scheme 4, both would receive equal credit. This lack of fairness can deter students from engaging in meaningful forum contributions.

Despite the above limitations, instructors have no choice but to rely on grading schemes like the ones discussed above. The large volumes of forum posts that accumulate by the end of the term make it impossible for the course staff to manually go through them to apportion credit. However, even if hypothetically, one were to have the course staff manually go through each of the contributions, there is a significant amount of subjectivity in assessing forum contributions. Having TAs manually grade contributions would lead to a lack of grading uniformity. A trivial contribution according to one TA, might be a significant contribution to another. Thus, there is a need for an automated way to assess the forum participation of students using a holistic grading scheme. Automation can lead to a standardized approach across the entire class.

The next sections discuss how we developed a system to assess forum participation by each student at scale. We go beyond the ready at hand statistics that are provided by the forum, and additionally incorporate measures that provide insight into the dynamics of students interacting in the forum. These dynamics manifest in the social networks that are created by the online interactions. We briefly review candidate predictors in the next section.

4. POTENTIAL PREDICTORS

The measurements of predictors arise from the data sets generated by forum facilities during the length of an academic term. Each offering of a course generates a separate data set, such as the one we used from the AI course.

Quantitative measures: These measures reward based on the volume of contributions made by an individual. As discussed in [24], higher forum participation count translates to higher learning gains, hence we include quantitative features in our list of potential predictors. These four predictors are: *number of questions asked*, *number of questions answered*, *total number of contributions*, and *average post length* by a student.

Engagement level measures: In order to reward the students who started important or intriguing threads, which in turn engaged many students, the *average number of collaborators* in the threads started by the student was added as a predictor. A second predictor, *average number of views* received by a student's questions was added for similar reasons. Given that not everyone in the class might be comfortable actively posting on the forum, we use some metrics to reward the passive engagement of the students. Some of the students are great listeners; they view or follow most of the posts, and are regularly online on the forum, which translates to passive forum participation. The two predictors we used to apportion credit for passive collaboration on the forum are: *total number of days a student was logged into the forum*, and the number of *posts viewed* by the student.

Quality measures: These measures are used to reward the students based on the quality of their contributions. These include upvotes and endorsement counts available in forum datasets. Students can express appreciation for a post by adding an upvote to the contribution. Instructors can explicitly endorse answers provided by students, marking those answers as definitive. Upvotes and endorsements articulate human judgments, and can be thought of as crowdsourcing post quality assessment.

Another strength of quality measures is their robustness to student cheating by flooding the forum with meaningless threads to increase their contribution count. Our two quality predictors are: *number of endorsed answers by the student*, and *total number of endorsements*, including upvotes on the questions, answers and instructors' endorsements.

Social Network Analysis: As discussed in the Related work section, Social Network Analysis (SNA) provides insights to the student forum participation. A brief detour in the following section provides background for the measures we used for SNA.

In order to include the SNA component in our credit apportioning system, the following networks were extracted from the class forum dataset. In the definitions below, nodes represent students and instructors. Typed edges represent interactions that are possible in the forum. Link weights encode the number of such interactions between the link's nodes.

Upvotes network: An upvotes network is extracted, where an edge from student A to student B indicates that A upvoted B's content at least once, and the weight of the edge encodes the number of times A upvoted B's content.

Endorsement network: An endorsement network is extracted, where an edge from instructor A to student B indicates that A endorsed B's content at least once, and the weight of the edge encodes the number of times A endorsed B's content.

Combined upvotes and endorsement network: This construct is a union of the above two networks. An edge from A to B indicates that A either upvoted and/or endorsed B's content at least once, and the weight of the edge encodes the sum of the upvotes and endorsements.

Interaction network: This graph models the interaction that happened on the forum over the duration of the course. In the interaction network, an edge from A to B indicates that B responded at least once to a question that A posted.

We use these networks to derive our final two predictors: *degree centrality* in the interaction network, and *page rank* in the combined upvotes and endorsement network.

We calculate the degree centrality for every node in the interaction network. Degree centrality measures the number of links incident upon a node. Higher degree centrality of a student implies that the student answered questions or resolved doubts for a large number of students. On a high level, degree centrality in the interaction network translates to the "helpfulness" and "resourcefulness" of the student. It also captures the breadth of the student's course knowledge.

Page rank in the combined upvotes and endorsement network was used in order to capture importance in both upvotes and endorsement information using a single metric. Page rank can additionally help uncover influential or important students in the network. Their reach extends beyond their immediate neighbors, and is therefore not captured by the earlier described upvote/endorsement measures. The higher the page rank in the combined network, the more "influential" the student.

5. GROUND TRUTH COLLECTION

In order to evaluate how effective each of the above predictors is in informing credit apportioning, we obtained human judgments by paying former students and teaching assistants of the AI or a related class to render judgments over a sample of posts. Given the high course enrollment of 700+, not all the posts could be evaluated. A survey instrument was used to collect the judgments, and participants were paid a $20 gift card. The number of posts sampled was limited by this cost, and time capacity of the 24 participants we could recruit.

Each item in the survey for the experts was a pair of two posts by different students. The experts were asked to indicate which of the two contributions was more helpful for the class as a whole. (See the precise instructions below). We chose this pairwise comparison method to economize on raters' time and attention, and because the derivation of full rankings from pairwise comparisons is well studied [11, 15].

Table 1: Kendall tau distance between rankings created by the 5 algorithms

	Algo1	Algo2	Algo3	Algo4	Algo5
Algo1	1	0.8538	0.2213	0.7243	0.2268
Algo2	0.8538	1	0.2132	0.6621	0.2050
Algo3	0.2213	0.2132	1	0.2306	0.3064
Algo4	0.7243	0.6621	0.2306	1	0.2741
Algo5	0.2268	0.2050	0.3064	0.2741	1

The task in preparing the survey was to find forum contribution pairs that would later help train an algorithm. The challenge was to select a set of posts that would cover a range of measures for all our candidate predictors, while being representative of the overall contributions. We describe here how this selection was accomplished.

Four algorithms use a weighted combination of the above explained candidate predictors.

- Alg 1: Using only quality measures and social network analysis measures.
- Alg 2: Using only quantitative measures and engagement level measures.
- Alg 3: Using all the measures with more emphasis placed on quantitative measures.
- Alg 4: Using all the measures with more emphasis placed on quality measures.

In addition, the current formula based grading scheme 1 that is used by some instructors at our university is included as a variant. Let us call this approach Alg 5:

$$\begin{aligned}\texttt{Score} = 1 * (\texttt{no\_questions\_asked}) + \\ 4 * (\texttt{no\_questions\_answered}) + \\ 0.5 * (\texttt{other\_contributions})\end{aligned} \tag{3}$$

All the above five algorithms are then separately used to calculate each student's score. Table 1 shows the Kendall tau distance between the rankings created by each of the algorithms. Most of the values in the table are low, indicating low correlation between the rankings calculated by each of the 5 algorithms. This result is intuitive because all the 5 algorithms were designed by us to capture slightly different signals. As a next step, 10 new values are calculated, each of which are absolute values of ranking differences between one pair of rankings for the same student. Each algorithm pair is processed. Thus, we have Alg1vsAlg2, Alg1vsAlg3, Alg1vsAlg4, Alg1vsAlg5 and so on. For instance, if Student ID# 500 was ranked 30 by Alg 1, and 300 by Alg 3, then the Alg1vsAlg3 value for Student ID# 500 would be 270.

We then sort these ten rank differences in descending order. The top entry in the 10 columns gives us the corner cases, or students 'of interest'. To sample student pairs, we compare these students of interest with the students immediately above and immediately below in the ranking by both the algorithm rankings under consideration. We clarify the procedure with the following example:

Let us assume that student ID #10 had the maximum absolute difference between ranking through Alg 1 and Alg 3. Also, using Alg 1, student ID #400 is directly above student ID #10 and student ID #5 is directly below student ID #10 in the ranking. Finally, let us assume that using Alg 3, student ID #20 is directly above student ID #10 and student ID #557 is directly below student ID #10 in the ranking. Then posts by the student ID pairs of interest for which human judgment was solicited are: (10, 400), (10, 5), (10, 20), (10, 557).

In addition to 40 such pairs of interest, additional student pairs were randomly sampled. At most 4 question pairs and 4 answer pairs were sampled from all these selected student pairs and presented to the experts. A total of 89 question pairs and answer pairs were used. In order to avoid fatiguing the experts, the set was partitioned into two batches such that each question pair or answer pair was voted on by at least 12 experts. The set of judged samples thus served to inform boundary cases among available measures, rather than to include every type of post. For example, there was no attempt to cover all linguistic variations. The addition of randomly sampled posts served to reach beyond this focus.

The survey instructions were as follows:
Each of the following sections presents one pair of questions or answers that were posted to the course forum in the past. We ask that you to indicate for each pair, the contribution that might have been most helpful to the rest of the class.

One sample item from the survey is as follows:

Q1: I am very confused about alpha-beta pruning, as we do not have example code from lecture. When we say we prune certain leaf, what does it mean? Does it mean we do not store that choices?
Q2: To create our own label, must it been binary label {1,-1} or it can be multi-categories with labels of any number? Is the feature still word counts or can be anything?
Which of the above two questions contributes more to the class community?

Note that in all cases the experts who answered the surveys were different from the experts whose endorsements we counted when building the classifier.

In order to learn the experts' *intuition* about which of the predictors might be important in ranking students' forum contributions, the following related question was introduced in the ranking survey once at a random time, with a facility to drag the entries up and down to arrange the predictors in decreasing order of relevance:

Imagine you had the following statistics about forum contributions by all students at the end of the term. In your opinion, which statistics are important to evaluate the forum contributions of students to the class. Please drag the entries up and down to indicate their relative importance. The first entry would be the most important.

- *Number of questions asked by the student*
- *Number of questions answered by the student*
- *Total number of posts viewed by the student*
- *Total number of days the student was online on the forum*
- *Total number of endorsements received by the student*
- *Total number of Forum contributions by the student (including questions, answers, notes, follow-ups, etc.)*

Table 2: Experts' intuitions for relative ordering of indicator importance. Example: 57.1% of experts felt that the number of questions answered was the second-most important indicator.

Rank	Feature	%support
1	# of endorsements	60.7
2	# of questions answered	57.1
3	# of Forum contributions	46.4
4	# of questions asked	46.4
5	# of posts viewed	60.7
6	# of days online	64.2

Based on the majority vote for every rank, we arrive at a ranking order using the experts' intuition. This ranking was not used in any of the experiments below. The information just illustrates the 'gut' feeling by our raters. The results are summarized in Table 2. Rank 1 is the most important feature. The percentage of experts agreeing with each ranking is also included.

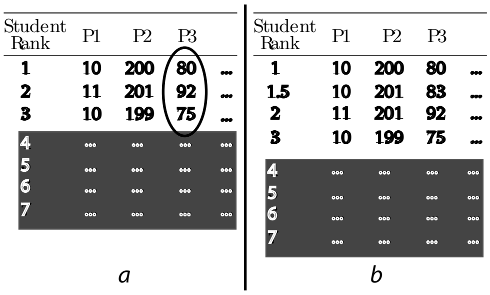
6. EXPERIMENT PREPARATION

Given the pairwise rankings of posts by the experts we needed to arrive at a ranking against which we could then train and test. We generated this ranking using the Copeland method [3]. The procedure counts the number of times a student's post was considered superior to the alternative post offered to an expert. The number of losses are then subtracted from these wins. Copeland ranking ties can be broken by a *second order Copeland* approach [5]. However, we found that forcing a complete order did not lead to good classification, because the ties are a reflection of true similarity.

We included at most 4 question pairs and 4 answer pairs from each sampled student in the survey. However, in most cases the sampled students had less than 4 questions / 4 answers. The final result was a list of 37 students for which we had rankings from twelve experts each. We collected this large number of rankings for each student because of the above mentioned subjectivity in evaluating posts. In addition to the rankings, we also had the measures for all our 12 candidate predictors for each of the 37 students .

Rather than attempting a regression, we formulated the problem as one of classification into four classes: the rank quartiles. This decision was based on the application of apportioning credit. A granularity of four suffices, given that forum participation is not the only source of credit for a course. Partitioning a 5% course credit into 700 values is not meaningful.

Given the sparsity of our human labeled set, we first augmented the labeled data as follows. We partitioned the ranked list of students into four roughly equal parts. Figure 4*a* shows the top two partitions using fictitious numbers for clarity.



Student Rank	P1	P2	P3	
1	10	200	80	…
2	11	201	92	…
3	10	199	75	…
4	…	…	…	…
5	…	…	…	…
6	…	…	…	…
7	…	…	…	…

a

Student Rank	P1	P2	P3	
1	10	200	80	…
1.5	10	201	83	…
2	11	201	92	…
3	10	199	75	…
4	…	…	…	…
5	…	…	…	…
6	…	…	…	…
7	…	…	…	…

b

Figure 4: Augmentation occurs separately within each quantile. Each column holds the measures of one predictor P_n. The top two quantiles are shown. Part a: before augmenting the top quantile; Part b: after augmentation.

We then determined the range of values for each predictor within one quartile. Finally, we generated new rows within each quartile by randomly choosing values for each of its predictors from within the range of values that the predictor took on within that quartile.

The four quantiles could not be filled equally because of the ranking ties. Tied students should be in the same class, rather than being split across quantile boundaries. When such splitting occurred we moved all participants into one of the quantiles, such that the fewest moves were required. For example, if three of five students with rank seven were assigned to quartile two, and two were assigned to quartile three, all students ranked seven were moved to quartile two.

Finally, we set aside 30% of the resulting augmented set for testing. We call these sets *forumTrainAug* and *forumTestAug*. The corresponding putative responses are *forumTrainResp* and *forumTestResp*. Our first exploration was to see whether we could construct a classifier that would use predictor measures to assign each student to one of the quartiles.

7. EXPERIMENT 1: QUANTILE PREDICTION USING RANDOM FOREST

We started with a random forest (RF) of 10K trees in order to understand how many trees are required for this classification. Figure 5 shows the result of this investigation.

Table 3: Accuracy and Kappa by number of predictors per tree

mtry	Accuracy	Kappa
2	0.89	0.85
7	0.88	0.85
12	0.88	0.84

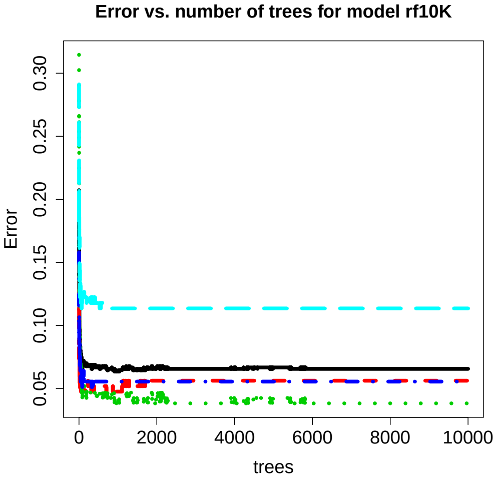


Figure 5: Classification errors by number of trees.

Each of the colored traces corresponds to one classifier. There are four traces, one corresponding to each quartile. The black line is the out-of-bag error. We see that after 6K trees the classification error no longer fluctuates. We settled on 8K trees to handle high data fluctuations. The second hyper parameter to tune, *mtry*, is the number of randomly chosen predictors that are used in each tree. The setting *mtry* == 2 was optimal, although this parameter is robust; see Table 3.

The resulting model *rf8K*, trained on *forumTrainAug* with 10-fold cross validation repeated 3 times has the confusion matrix shown in Table 4.

Table 4: Model RF8K predicting 308 augmented test set outcomes. Accuracy: 0.94

	RefQ1	RefQ2	RefQ3	Ref4	Class Error
PredQ1	76	0	2	0	0.03
PredQ2	1	77	0	14	0.16
PredQ3	0	0	75	0	0.00
PredQ4	1	0	1	62	0.02

Figure 6 shows the relative importance of our candidate predictors.

The chart shows the amount of decrease in accuracy that is contributed by each of the predictors. The top three predictors are the number of student answers that were endorsed by an instructor, the total number of endorsements, and the number of days the student was online on the forum. Note that these predictors differ somewhat from those intuited by the experts, though there is some overlap.

Since there some of the predictors are partially covariant We experimented using three predictors only, but the degradation was noticeable. It is also advantageous to retain predictors that are less easy to spam than time online. For instance, the page rank predictor, while less important for the classification, is more difficult to defraud.

Using *rf8K* we predicted *forumTestResp*. Figure 7 shows ROC curves for each quartile predictor.

The prediction accuracy reaches 0.96. This result is encouraging in that it signals inroads towards apportioning fair forum participation credit even for very large courses.

However, the result does not speak to generality. The model was trained on a science forum data set, and its human labels were few. The classifier would not be useful if new labels needed to be created for each class. We therefore added a second experiment to demonstrate how the approach behaves when training occurs on data of an unrelated domain, and the resulting classifier is then used to predict forum participation ranking.

8. EXPERIMENT 2: STACK EXCHANGE TRANSFER LEARNING

Constructive activity on the Stack Exchange [2] forum earns users *reputation*, which can be used as a surrogate for forum participation credit. Among others, measures similar to the Piazza statistics we used in Experiment 1 are available from Stack Exchange, and we used those to predict reputation. However, only one of these measures is used by Stack Exchange for *their* computation of reputation; SE's algorithm instead takes six other variables into account.

We obtained the Stack Exchange (SE) records for the site dedicated to Economics [2].

We began with the data from about 5300 SE contributors. In a first step we followed the same procedure as in Experiment 1 to obtain optimal *mtry* and forest size values, which were 2 and 4K respectively. After scaling, centering, and partitioning into quartiles we set aside a 30% test set (*seTest*) from the training set (*seTrain*). The respective reputation responses are *seTrainResp*, and *seTestResp*.

Since the forum training set was not involved in the SE training, we used the larger *forumTrainResp* as test target for the SE-trained forest. Figure 8 shows the problematic resulting AUC ROC curves.

We addressed the lower triangle $Q3$ curve by reversing that classifier's orientation. This step is an appropriate measure, because the curve lies consistently below the diagonal, indicating a true polarity issue. However, AUC values were low, and further investigation uncovered the reason (Figure 9).

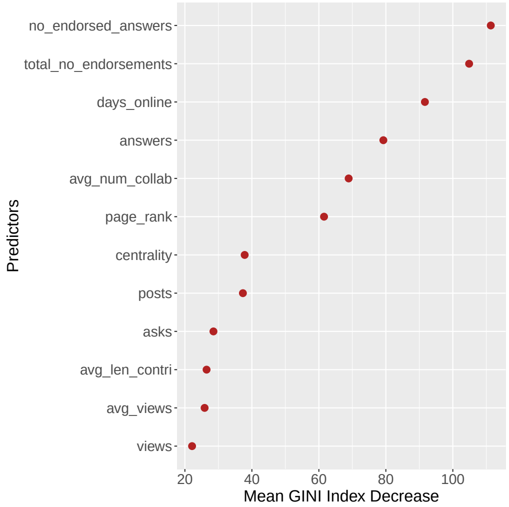


Figure 6: Mean decrease in GINI (node purity) when removing individual predictors. Ordered from most important at the top to least important.

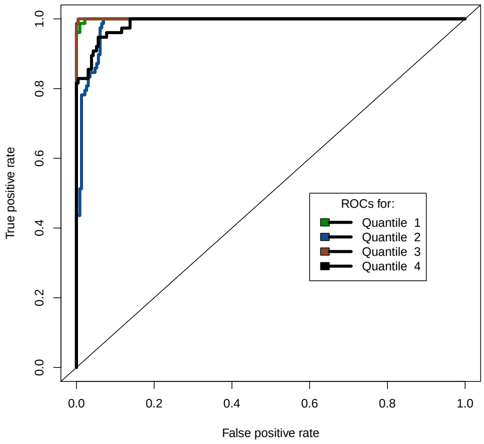


Figure 7: ROC curves for each quartile, predicted by 8000 random forest trees.

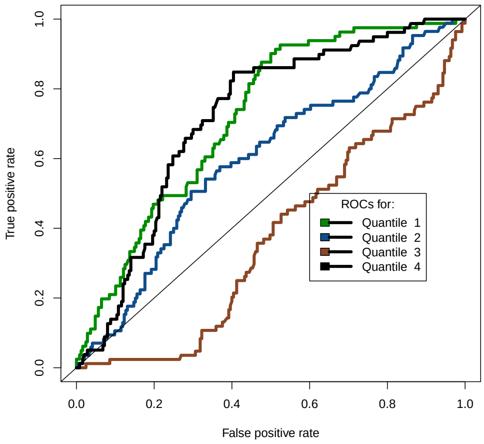


Figure 8: Initial AUC ROC from Stack Exchange-trained random forest predicting forum contribution quality.

Table 5: Confusion matrix for RF4K. OOB estimate of error rate: 27.81%

	Q1	Q2	Q3	Q4	Class error
Q1	502	25	83	313	0.46
Q2	3	859	1	34	0.04
Q3	149	6	716	29	0.20
Q4	126	230	9	539	0.40

Table 6: AUC Stack Exchange-trained model predicting forum post quality

	Q1	Q2	Q3	Q4	Mean
forumTrainResp	0.72	0.62	0.64	0.76	0.69
forumTestResp	0.78	0.64	0.45	0.77	0.66

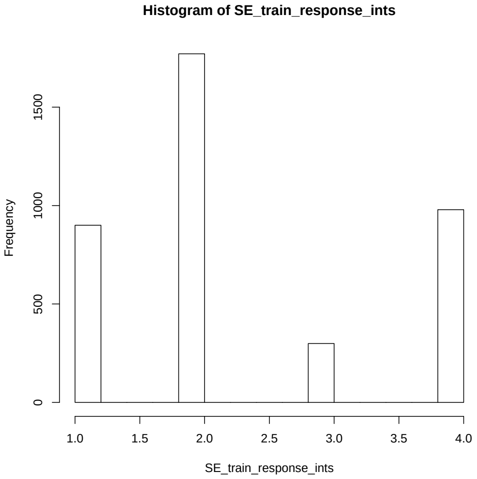


Figure 9: Class imbalance with raw Stack Exchange data

The Figure 9 shows that quartile 2 is over-represented, while quartile 3 suffers from a scarcity of examples. We balanced the training set by subsampling the quartile 2 examples to 1200, and augmented quartile 3 examples analogously to our process in Experiment 1.

The resulting 4K tree model, again trained with 10-fold cross validation repeated three times yielded a training accuracy of 0.72, and a kappa of 0.63. Table 5 shows the model's confusion matrix. When predicting *seTest* with this SE-trained classifier, a satisfactory mean AUC of 0.93 resulted, with classification behaviors shown in Figure 10.

Finally, with the SE model reasonably solid, we used this model to once again predict both *forumTrainResp* and *forumTestResp*. Table 6 shows results.

An important question remains: how do the ad hoc formulas devised by instructors perform? Are they sufficient? A final experiment tested the power of the informally designed Scheme 1 to approach the human expert judgments. Experiment 3 examines this question.

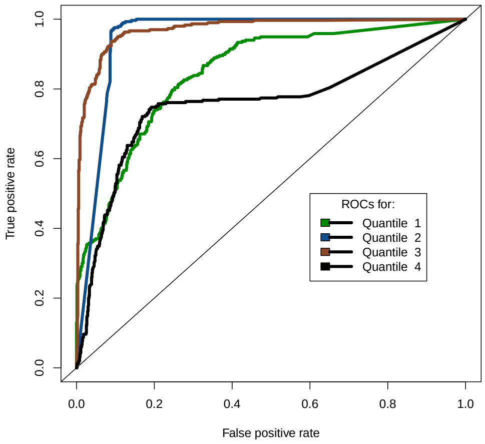


Figure 10: ROC for predicting Stack Exchange reputation from Stack Exchange-trained 4K random forest after attending to class imbalance.

9. EXPERIMENT 3: COMPARISON WITH CURRENT PRACTICE

We computed the quartile predictions induced by Equation 1, and compared them against $forumTestAug$ using Cohen's Kappa. This test returned a value of zero, evidence that the equation does not generate the same quartile assignments as the human experts. As a final check, we produced the categorical 1/0 quartiles for $forumTestAug$ from the $rf8K$ model, using 0.5 as the probability cutoff. The Cohen's Kappa between our model's prediction and the experts was 0.94.

10. DISCUSSION

The average AUC of 0.66 when using the Stack Exchange trained classifier on forum posts lags behind the classifier that is specialized on forum post evaluation. However, as a first step this result is encouraging. Forum assessment is gaining enough importance, and human judgments are expensive enough that training data from large, ready at hand, and similar enough facilities is extremely attractive for attempts in transfer learning.

Stack Exchange and other reputation incentivized systems have accumulated enough labeled samples that alternatives to random forests, such as neural nets, which require large amounts of training data might be feasible as approaches going forward.

11. CONCLUSION

Forum assessment is an active research area for good reason. A growing number of schools and companies are offering entire degree programs online, all of which require online communication among students and instructors. Demand for tools that help manage and assess forum activity is likely to rise as online education continues to capture market share.

Given that our attempt at transfer learning worked reasonably well, exploring the use of neural networks for automatic forum participation grading is our next step. In addition, the work described here has not yet leveraged the content of the forum posts in assessing forum participation. In [26], the authors show that computational linguistic models can help in measuring learner motivation and cognitive engagement from the text of the forum posts. Hence, we plan to leverage Natural Language Processing techniques to analyze the content of the posts, and use those in apportioning forum participation credit. As explained in the introduction, this work is part of a larger effort that fills modules into a forum centered architecture. The frequently asked questions module and spam detection module will round out our efforts going forward.

12. REFERENCES

[1] Economics beta. World-Wide Web. Accessed Mar 6, 2018.

[2] Stack exchange data dump. World-Wide Web, 12 2017.

[3] A.H.Copeland. A âĂŸreasonableâĂŹ social welfare function. Notes from a seminar on applications of mathematics to the social sciences., 1951.

[4] M. A. Andresen. Asynchronous discussion forums: success factors, outcomes, assessments, and limitations. *Journal of Educational Technology & Society*, 12(1):249, 2009.

[5] W. contributors. Copeland's method — wikipedia, the free encyclopedia, 2016. [Online; accessed 7-March-2018].

[6] M. De Laat, V. Lally, L. Lipponen, and R.-J. Simons. 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):87–103, 2007.

[7] B. De Wever, T. Schellens, M. Valcke, and H. Van Keer. Content analysis schemes to analyze transcripts of online asynchronous discussion groups: A review. *Computers & education*, 46(1):6–28, 2006.

[8] N. M. Dowell, O. Skrypnyk, S. Joksimovic, A. C. Graesser, S. Dawson, D. GaĹĚĞevic, T. A. Hennis, P. de Vries, and V. Kovanovic. Modeling learners' social centrality and performance through language and discourse. *International Educational Data Mining Society*, 2015.

[9] D. R. Garrison, T. Anderson, and W. Archer. Critical thinking, cognitive presence, and computer conferencing in distance education. *American Journal of distance education*, 15(1):7–23, 2001.

[10] A. A. Gokhale. Collaborative learning enhances critical thinking. *Journal of Technology Education*, 1995.

[11] R. Heckel, M. Simchowitz, K. Ramchandran, and M. J. Wainwright. Approximate ranking from pairwise comparisons. *CoRR*, abs/1801.01253, 2018.

[12] F. Henri. Computer conferencing and content analysis. In *Collaborative learning through computer conferencing*, pages 117–136. Springer, 1992.

[13] C. Howell-Richardson and H. Mellar. A methodology for the analysis of patterns of participation within computer mediated communication courses. *Instructional Science*, 24(1):47–69, 1996.

[14] 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, New York, NY, USA, 2014. ACM.

[15] K. G. Jamieson and R. D. Nowak. Active ranking using pairwise comparisons. *CoRR*, abs/1109.3701, 2011.

[16] D. Laurillard. *Rethinking university teaching: A conversational framework for the effective use of learning technologies.* Routledge, 2013.

[17] R. M. Marra, J. L. Moore, and A. K. Klimczak. Content analysis of online discussion forums: A comparative analysis of protocols. *Educational Technology Research and Development*, 52(2):23, 2004.

[18] D. Nandi, S. Chang, and S. Balbo. A conceptual framework for assessing interaction quality in online discussion forums. *Same places, different spaces. Proceedings ascilite Auckland*, pages 7–23, 2009.

[19] D. R. Newman, B. Webb, and C. Cochrane. A content analysis method to measure critical thinking in face-to-face and computer supported group learning. *Interpersonal Computing and Technology*, 3(2):56–77, 1995.

[20] L. F. Pendry and J. Salvatore. Individual and social benefits of online discussion forums. *Computers in Human Behavior*, 50:211–220, 2015.

[21] R. Rabbany, S. ElAtia, M. Takaffoli, and O. R. Zaïane. Collaborative learning of students in online discussion forums: A social network analysis perspective. In *Educational data mining*, pages 441–466. Springer, 2014.

[22] J. Scott and P. J. Carrington. *The SAGE handbook of social network analysis.* SAGE publications, 2011.

[23] L. A. Tomei, editor. *Matthew Shaul: Information Communication Technologies for Enhanced Education and Learning: Advanced Applications and Developments: Advanced Applications and Developments*, chapter Assessing online discussion forum participation. IGI Global, 2008.

[24] X. Wang, D. Yang, M. Wen, K. Koedinger, and C. P. Rosé. Investigating how student's cognitive behavior in mooc discussion forums affect learning gains. *International Educational Data Mining Society*, 2015.

[25] S. Wasserman and K. Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

[26] M. Wen, D. Yang, and C. P. Rosé. Linguistic reflections of student engagement in massive open online courses. In *ICWSM*, 2014.

[27] Wikipedia. *Piazza*, 2017.

[28] H.-T. Yeh. The use of instructor's feedback and grading in enhancing students' participation in asynchronous online discussion. In *Advanced Learning Technologies, 2005. ICALT 2005. Fifth IEEE International Conference on*, pages 837–839. IEEE, 2005.

[29] N. Yusof, A. A. Rahman, et al. Students' interactions in online asynchronous discussion forum: A social network analysis. In *Education Technology and Computer, 2009. ICETC'09. International Conference on*, pages 25–29. IEEE, 2009.

An Open Vocabulary Approach for Estimating Teacher Use of Authentic Questions in Classroom Discourse

Connor Cook
Institute of Cognitive Science
University of Colorado Boulder
Boulder, CO 80309
connor.cook@colorado.edu

Andrew M. Olney
Institute for Intelligent Systems
University of Memphis
Memphis, TN 38152
aolney@memphis.edu

Sean Kelly
University of Pittsburgh
Pittsburgh, PA 15260
spkelly@pitt.edu

Sidney K. D'Mello
Institute of Cognitive Science
University of Colorado Boulder
Boulder, CO 80309
sidney.dmello@colorado.edu

ABSTRACT

Automatic assessment of the quality of classroom discourse can have a transformative effect on research and practice on improving teaching effectiveness. We improve on a previous automated method to measure teacher authentic questions – open-ended questions without pre-scripted responses that predict student achievement growth – using classroom audio and expert question codes from two sources: (1) a large archival database of text transcripts of 428 class-sessions from 116 classrooms, and (2) a newly collected sample of 132 high-quality audio recordings with automatic speech recognition transcripts from 27 classrooms. Whereas previous work utilized a "closed vocabulary" approach, consisting of 732 pre-defined word, sentence, and discourse level features, the present "open vocabulary" approach exclusively utilized word and phrase counts from the transcripts themselves. The two approaches yielded substantial, but statistically equivalent, correlations with gold-standard human codes of authenticity (Pearson *r*'s of 0.396 vs. 0.424 and 0.602 vs. 0.613 for datasets 1 and 2, respectively). Importantly, averaging estimates from the two approaches resulted in statistically significant improvements over either approach (*r's* of 0.492 and 0.686 for datasets 1 and 2, respectively). We discuss implications of our findings for automated analysis of classroom discourse.

Keywords

Open vocabulary, authentic questions, classroom discourse

1. INTRODUCTION

(Example 1)

Teacher*: "How does a person become a noble?"*

Student: "*They're born into it."*

Teacher: *"They're born into it, right? It's by family. It gets passed down so if you're a noble, your child would be a noble, their child would be…it's a tradition, right?"*

(Example 2)

Teacher: "*How did that make you guys feel, I mean what was your gut reaction to all that?"*

Student*: "Ashamed."*

Teacher: "*Ashamed in what way?"*

Consider these discourse exchanges between a teacher and his/her students from an actual classroom. The first follows the oft-used, but ineffective, Initiate-Response-Evaluate (IRE) [40] mode of questioning. Now contrast this with the second case, where the teacher asks an open-ended question or a question without a pre-scripted response. Although it only elicited a one-word answer from the student, the teacher withheld evaluation, instead building on the student's response, thereby "opening up" the conversation.

Such questions – called *authentic questions* — whose answers are not presupposed by the teacher (e.g. "Do you think Abigail is going to tell the truth?" [33]) are a core dimension of dialogic instruction related to student engagement and achievement growth [24, 25, 42], and are central to many conceptual models of effective discourse practices [39, 50, 63]. Prior research utilized expert human coders to identify discourse practices at the level of individual questions and thus provided exceptionally precise measures of instructional practice. Our goal is to precisely estimate the prevalence rate of teacher authentic questions using fully-automated methods.

Why bother in the first place? It is because teacher observation has become increasingly central to educational research and school improvement efforts [2, 26, 28, 35, 58]. Observations of classroom practice are valuable because they identify specific domains of practice for improvement [36] and can target dimensions of schooling not captured by test scores, such as socialization processes in elementary school [32]. Classroom observations also enhance school principals' role in managing teachers' work [30]. Yet current in-person observational methods are logistically complex, require observer training, are an expensive allocation of administrators' time [4], and simply do not scale.

Can computers help? We think so, and report the results of ongoing research efforts to automate the analysis of teacher question-asking behavior, a common component across various well-known observation protocols (e.g., Domain 3 of Danielson's Framework for Teaching [16]; PLATO's Classroom Discourse Element [27]). Our specific emphasis on authentic questions is motivated by the strong research base linking them to engagement and achievement as cited above.

1.1 Related Work

There has been considerable work on detecting questions from text [1], with fewer studies focusing on audio [8, 45, 61]. These studies also largely focus on general question detection from meetings and other interactions, which is quite different from the present goal of detecting authentic questions from real-world classrooms. Blanchard et al. [6] and Donnelly et al. [20] investigated question detection from classroom audio, but again, their emphasis was on discriminating questions from other utterances, which is a related but distinct problem from authenticity detection. There has also been research on automated analysis of teacher and student discourse [18, 19, 62], but these studies emphasize modeling of general instructional activities (e.g., distinguishing between lecture vs. group work vs. discussion) rather than authentic questions.

To our knowledge, there have only been three studies germane to our goal of detecting authentic questions from classroom discourse. Samei et al. [53] focused on identifying authenticity from human-transcribed questions from the Partnership for Literacy Study, a large sample of over 20,000 questions and associated "gold-standard" human codes (see section 2.1). The authors repurposed features (e.g., part of speech tags) from an existing speech act classifier [44] to train a J48 classifier to detect authenticity of individual questions. They achieved a Cohen's kappa of 0.34 and accuracy of 67%, which they deemed promising but in need of improvement.

In a follow-up study, Samei et al. [54] focused on testing the generalizability of this model. They split the data based on whether it was collected in an urban or non-urban area and whether the teacher had been trained in dialogic practices (including the use of authentic questions and other effective teacher talk strategies). They found that classifiers trained on a subset (e.g. urban) and tested on the dual subset (e.g. non-urban) were fairly close in accuracy to one another, but that some subpopulations were more representative of the data than others, making them better for classifier training.

Of utmost relevance to the present study is work by Olney et al. [43] on detecting authentic questions from the aforementioned Partnership dataset as well as a newly collected CLASS 5 dataset with automatic speech recognition (ASR) transcriptions (see Section 2.1). Their main goal was to address heavily imbalanced classes, which occur because of the relatively infrequent proportion of authentic questions (about 3%) compared to all teacher utterances. The class imbalance problem was so severe that they forewent identification of individual authentic questions, instead focusing on predicting the proportion of all utterances in a class session that were authentic questions. In other words, an utterance-level binary prediction problem (i.e., labeling an utterance as an authentic question or not) was recast as the problem of predicting the proportion of authentic questions at the class level.

Using a combination of 242 pre-defined features, extracted at the word, sentence, and discourse level, they first attempted aggregating utterance-level predictions of authentic questions, obtained with SMOTEBoost [11], to the class level. This yielded correlations of 0.27 and 0.44 between the predicted and actual (human-coded) authenticity proportions on the Class 5 and Partnership datasets, respectively. The difference in correlations was attributed to the differences in the degree of class imbalance across the two datasets because the Partnership data only contained instructional questions whereas the Class 5 data contained all teacher utterances. Next, they aggregated their utterance-level features to the class level (by taking their mean, sum, and standard deviation to yield 726 features) and then trained a M5P regression tree [23] on the resulting class-level features. The resulting correlation increased from 0.27 to 0.50 for the Class 5 dataset (with the most severe imbalance) but remained similar (0.42 vs. 0.44) for the Partnership dataset (with minor imbalance). Further refinements by Kelly et al. [37], including adding 6 new class-level features, resulted in correlations of 0.61 and 0.42 on the Class 5 and Partnership datasets, respectively.

We attempt to improve on these results using an open vocabulary approach for class-level authenticity prediction. In an open vocabulary approach, the features used to train a classifier are determined from the data itself and are not pre-determined. To illustrate, albeit in a different domain, Schwartz et al. [56] used an open vocabulary approach to predict gender, age, and personality traits based on social media posts. They computed counts of words and phrases (i.e., n-grams) per participant, and then filtered phrases based on pointwise mutual information (PMI) [13, 38], which ensured that they only kept phrases with high informational value. They then normalized the word and phrase counts by the total number of words for each participant and applied the Anscombe transformation [3] to the normalized values to stabilize their variances. They also generated topics using Latent Dirichlet Allocation (LDA) [7, 59]. Using words, phrases, and topics as features, the authors were able to predict gender, age, and personality traits more accurately than a closed vocabulary approach using features from Linguistic Inquiry and Word Count (LIWC) [48, 49]. We apply a variant of this basic approach in the present study.

1.2 Novelty and Contributions

We expand on and improve upon previous work [43] on automatically estimating the proportion of authenticity in classroom discourse using the same datasets. We call this previous approach a closed vocabulary approach since the features are predefined and are independent of the dataset. An advantage of the closed vocabulary approach is that it is less likely to overfit to the dataset at hand because it does not directly encode (as features) specific words from the corpus. This might be particularly important in the case of classroom discourse because generalizable models should encode language that correlates with authentic questions vs. being specific to the particular topic being discussed in class (e.g., The American Civil War).

In contrast, an open vocabulary approach uses counts of words and phrases found in the corpus. The vocabulary is "open" in that the features change depending on the corpus. A potential disadvantage of this approach is that it is more likely to overfit to the training dataset. However, we think this problem can be alleviated by careful selection of words and phrases for use as features. The advantage of this approach is that it ostensibly allows for the detection of a wider variety of instructional constructs due to a lack of pre-determined features. It also yields more interpretable models in that one can examine the specific words, phrases, and utterances that signal authenticity compared to some of the pre-defined features used in the closed vocabulary approach.

Previous research [56] has indicated that an open vocabulary approach outperforms the closed vocabulary approach on a different task of gender, age, and personality prediction from social media. How might it fare for the present task of authenticity prediction and what are the words and phrases that signal authenticity? Is there an advantage to combining both approaches? These are the questions that motivated the present study.

2. METHOD

2.1 Datasets

CLASS 5 (new) data. CLASS 5 data were collected between January 2014 and May 2016 from 132 classes taught by 14 different teachers at seven schools in rural Wisconsin. The data consisted of in-class observations in the form of live coding of authenticity by trained researchers and subsequent offline refinement of the coding from recorded audio. Both teacher and school identifiers were preserved with the data.

Given the logistical constraints of using individual microphones for each student, the recording instrumentation instead focused on high-quality teacher audio suitable for ASR (see [15] for a description of the setup). Classroom audio, which included both teacher and student speech, was recorded from a stationary boundary microphone, and is not of sufficient quality to be used for ASR; it is useful for marking when students speak but is not analyzed further here. Thus this dataset differs from the archival data (see below) in that the audio is automatically segmented into utterances, which are converted into transcripts using Bing Speech ASR with accompanying errors. Further, only teacher speech is transcribed, and the transcripts contain all utterances rather than just questions.

Partnership (archival) data. The archival data was collected in the Partnership for Literacy Study (Partnership), a study of professional development, instruction, and literacy outcomes in middle school English and language arts classrooms. The study collected data from 7th- and 8th- grade English and language arts teachers in Wisconsin and New York State from 2001 to 2003. Over that two-year period, 119 classrooms in 21 schools were observed twice in the fall and twice in the spring. Three of the classrooms had missing question data and could not be used for this study, leaving us with 116 classrooms. Classroom observations for Partnership were conducted using a near-real-time computer-based annotation system [41]. The primary focus of the system was to annotate the dialogic properties of questions asked by both teachers and students. During this process, the instructional questions were transcribed by humans, and the transcriptions were mostly accurate, but not verbatim. Reliability studies indicate that raters agree on question properties approximately 80% of the time, with observation-level inter-rater correlations averaging approximately .95 [42].

Table 1 shows a comparison of both datasets. Note that the same rubric was used to code authentic questions in both datasets.

2.2 Natural Language Processing

Closed vocabulary approach. The closed vocabulary approach used 732 specific features to predict the proportion of authentic questions in class sessions. This feature set includes specific words (like “Why” and “What”), part-of-speech tags, named entity type categorizations (such as PERSON, LOCATION, and DATE), syntactic dependencies (like subject, direct object, and indirect object), and discourse-level features (such as contrast and elaboration discourse relations, and joint, nucleus, and satellite elementary discourse units). There were 242 utterance-level features, which were aggregated at the class level by taking their mean, sum, and standard deviation [43]. Two more features were later added at the utterance level, leading to six more features at the class level, for a total of 732 class-level features [37].

Open vocabulary approach. The open vocabulary approach used a variable number of features depending on the dataset. This method was adapted from the open vocabulary language model developed by Park et al. [46]. To start, counts of words, two-word phrases, and three-word phrases were computed from the corpus. See Table 1 for a comparison of n-gram counts prior to filtering (see below).

We used a stop word list from Pedregosa et al. [47] to filter out the most common English words (such as “the” and “and”), and so these words and phrases including them were filtered out. We also required each word or phrase to occur in at least some percentage of documents, which we call the *cutoff* (we investigated multiple cutoffs, with results shown in Section 3).

We then calculated the pointwise mutual information (PMI) of each phrase, defined as:

$$pmi(phrase) = \log(\frac{p(phrase)}{\Pi\, p(word)})$$

where $p(phrase)$ is the probability of a phrase based on its relative frequency in the training data and $\Pi\, p(word)$ is the product of the probabilities of each word in the phrase in the training data. We filtered out phrases where the PMI was less than three times the number of words in the phrase [13, 38]. This helped ensure that we only used meaningful phrases (such as “language arts”), rather than phrases that were just the result of frequent words occurring next to one another (such as “next we will”). We experimented with PMI thresholds ranging from zero to four times the number of words in the phrase, but no difference in performance was observed. Cutoff and PMI filtering were based only on data in the training folds, ensuring that the test was not affected (see Section 2.3).

Combined approach. We simply averaged predictions from the closed and open vocabulary approaches.

Table 1. Summary of the two datasets

Item	Class 5	Partnership
# Utterances	45,044	Unknown
# Instructional Questions	4,377	25,711
# Authentic Questions	1,510	12,862
% Authentic Utterances	3%	Unknown
% Authentic Questions	34%	50%
Unigrams	17,520	8,358
Bigrams	152,023	61,460
Trigrams	319,545	117,049

Note. % Authentic Utterances refers to teacher utterances aligned with authentic questions. % Authentic Questions refers to instructional questions that were also authentic. N-gram counts are prior to filtering.

2.3 Model Training

We used M5P model trees, which are decision trees that have regression functions at each leaf node [23]. Starting at the root of the tree, decisions to follow a left or right branch are based on the value of a particular feature until a leaf with the appropriate regression model is reached. We chose the M5P model to enable comparisons with previous work [43].

All models used cross-validation, with selection of words and phrases to use as features for the open vocabulary approach based only on the training folds; we did not peek into the testing folds. For generalizability to new teachers, it was important that a teacher would not appear in both the training and testing folds. For the CLASS 5 data, this was achieved using leave-one-teacher-out cross-validation. For the archival Partnership data, the mapping between teachers and data files was incomplete, and so the mapping between schools and data files was used instead. This leave-one-school-out cross-validation assumes that a teacher did not transfer between schools during the study (a likely assumption), and in a sense is even more conservative than leave-one-teacher-out because it controls for similarities shared by teachers at the same school.

It should be noted that the unit of analysis is always a class-session. That is, counts for the language model, feature aggregation, and authenticity aggregation are all done at the level of an individual class-session.

2.4 Method Pseudocode

Below is pseudocode outlining our method for teacher-level cross-validation.

```
Aggregate utterance-level transcripts to the class session level
For each cutoff percentage:
    For each teacher:
        Split data into training set (class sessions from other teachers) and
                        test set (class sessions from this teacher)
        Get counts of n-grams (words, bigrams, and trigrams) for each class session in training set
        Remove n-grams that contain words from stop word list
        Remove n-grams that appear less than once in cutoff percentage of class sessions
        Filter phrases (bigrams and trigrams) using pointwise mutual information
        Get counts of kept n-grams for each class session in test set
        Train M5P model on n-gram counts from training set class sessions
        Use M5P model to predict authenticity on test set class sessions
    Pool class session authenticity predictions across teachers
    Compute correlation between predicted and actual authenticities for cutoff percentage
```

3. RESULTS

Our outcome measure is the Pearson correlation between the computer- and human-coded estimates of proportion authenticity per class session. We recomputed the previous results [37] obtained with the closed vocabulary approach and replicated the previous findings.

3.1 Cutoff Percentage (Open Vocabulary Approach)

As mentioned in Section 2.2, we tested various cutoff percentages for the open vocabulary approach. As can be seen in Figure 1, the correlation starts out low as the model is overwhelmed by the sheer number of features (Figure 2). However, as the cutoff becomes more stringent and the number of features decreases, the results improve, until the correlations peaks at 0.602, achieved with 52 features at an 82% cutoff. Beyond this point, the correlation steeply drops as too few features remain.

We observed a different pattern for the Partnership data as noted in Figure 3 and Figure 4. Here, the results were less dependent on the number of features, though the best correlation of 0.396 was obtained at the 61% cutoff with only 6 features retained. It should be noted that we only considered up to a 70% cutoff for this dataset because there were only three remaining features beyond this point. This is unsurprising because the Partnership data, though more diverse, only contains questions compared to the full transcripts in the CLASS 5 dataset, and consequently contains far fewer unique n-grams (see Section 2.2).

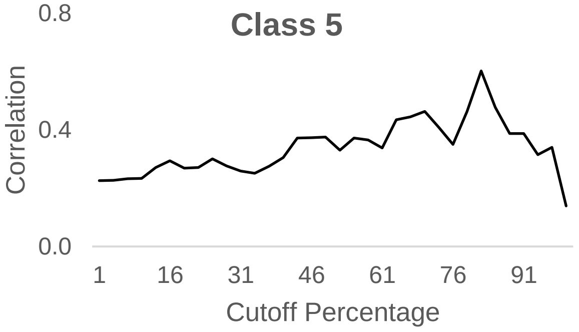


Figure 1. Correlation by cutoff % for Class 5 dataset

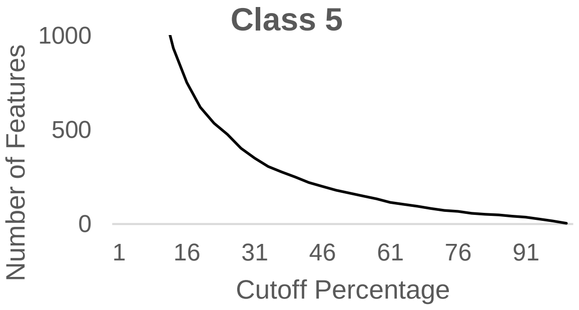


Figure 2. # of features by cutoff % for Class 5 dataset

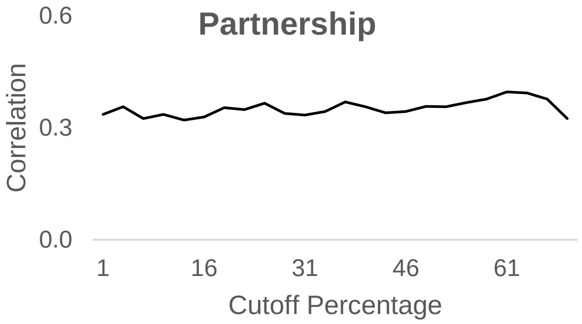


Figure 3. Correlation by cutoff % for the Partnership dataset

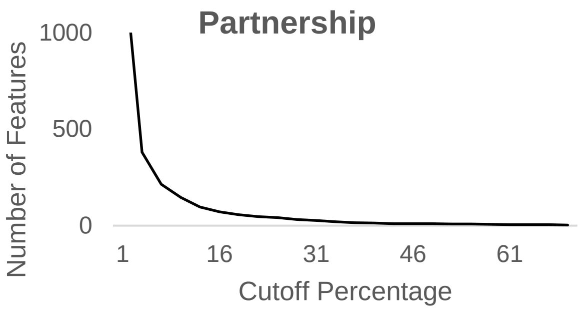


Figure 4. # of features by cutoff % for the Partnership dataset

3.2 Comparison with Closed Vocabulary Results

For the Class 5 data, the best correlation of 0.602 obtained via the open vocabulary approach was significant ($p < .001$) and similar to the significant 0.613 ($p < .001$) correlation obtained from the closed vocabulary approach. Zou's [66] test of the difference between two overlapping dependent correlations with one common variable (i.e., the gold-standard authenticity codes) indicated that the two correlation coefficients were statistically equivalent at the $p < .05$ level. A similar pattern of results was obtained for the Partnership data in that the significant 0.396 ($p < .001$) correlation from the open vocabulary approach was statistically equivalent to the 0.421 significant ($p < .001$) correlation from the closed vocabulary approach at the $p < .05$ level. Subsequent results focus on these two "best" models.

3.3 Combined Models

The analyses thus far indicate that the closed and open vocabulary approaches were equally predictive of authenticity across both datasets. Authenticity estimates from both methods correlated at .559 ($p < .001$) and .371 ($p < .001$) for the Class 5 and Partnership datasets, respectively, suggesting some, but not substantial, redundancy. This raises the question of whether a combination of the two approaches might improve predictive power.

We addressed this question by averaging the predictions of the two best models (we also attempted feature-level fusion, but this resulted in lower performance; results not shown here). For Class 5, the combined model predicted authenticity with a significant correlation of .686 ($p < .091$), which was quantitatively and statistically higher ($p < .05$) than the 0.602 and 0.613 correlations obtained from the open and closed vocabulary approaches, respectively (see Figure 5).

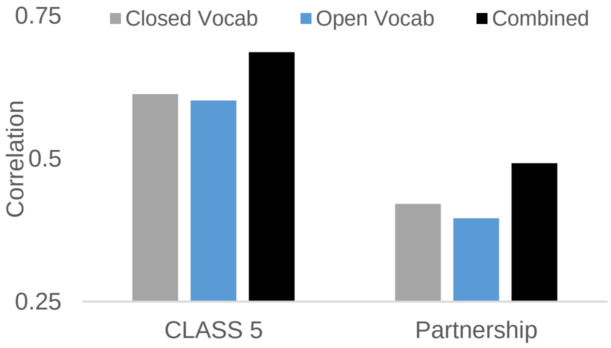


Figure 5. Comparison of closed, open, and combined models

These results can be visualized as a density plot (see left of Figure 6). The plot illustrates smoothed histograms of class-level computer- and human-provided proportional authenticity estimates. We note the combined model tends to slightly overestimate the mean compared to the human-coded data. Its predictions are also less positively skewed, ostensibly because it underpredicts some cases with considerable human-coded authenticity (also see right of Figure 6).

A similar pattern of results was obtained for the Partnership data. Specifically, the combined model's correlation of .492 was significant ($p < .001$) and also significantly higher ($p < .05$) than the 0.396 and 0.421 correlations obtained from the open and closed vocabulary approaches, respectively (see Figure 5). As noted in the density plot in Figure 7, the combined model is "peakier" with a reduced range in either direction compared to the human-coded data. The model has difficulty with cases associated with very low and very high human-coded authenticity (see scatterplot in Figure 7).

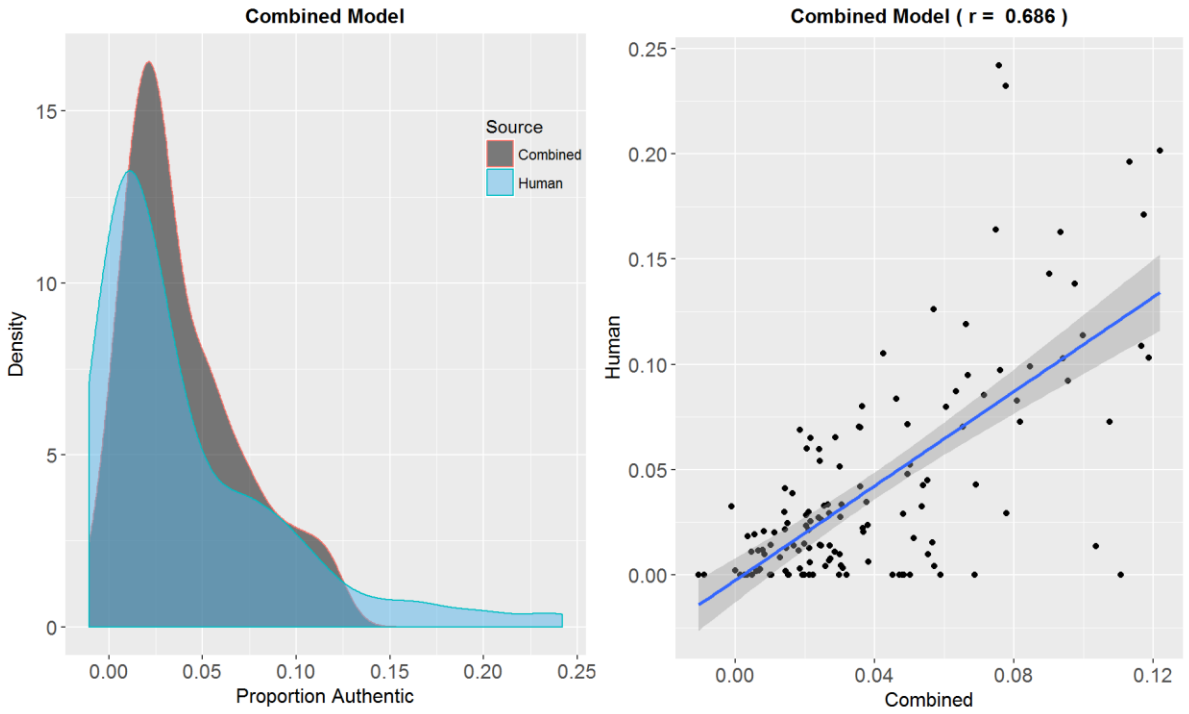


Figure 6. Density plot and scatter plot showing the resulting predictions from combining both the open and closed vocabulary models on the Class 5 dataset compared to human codes.

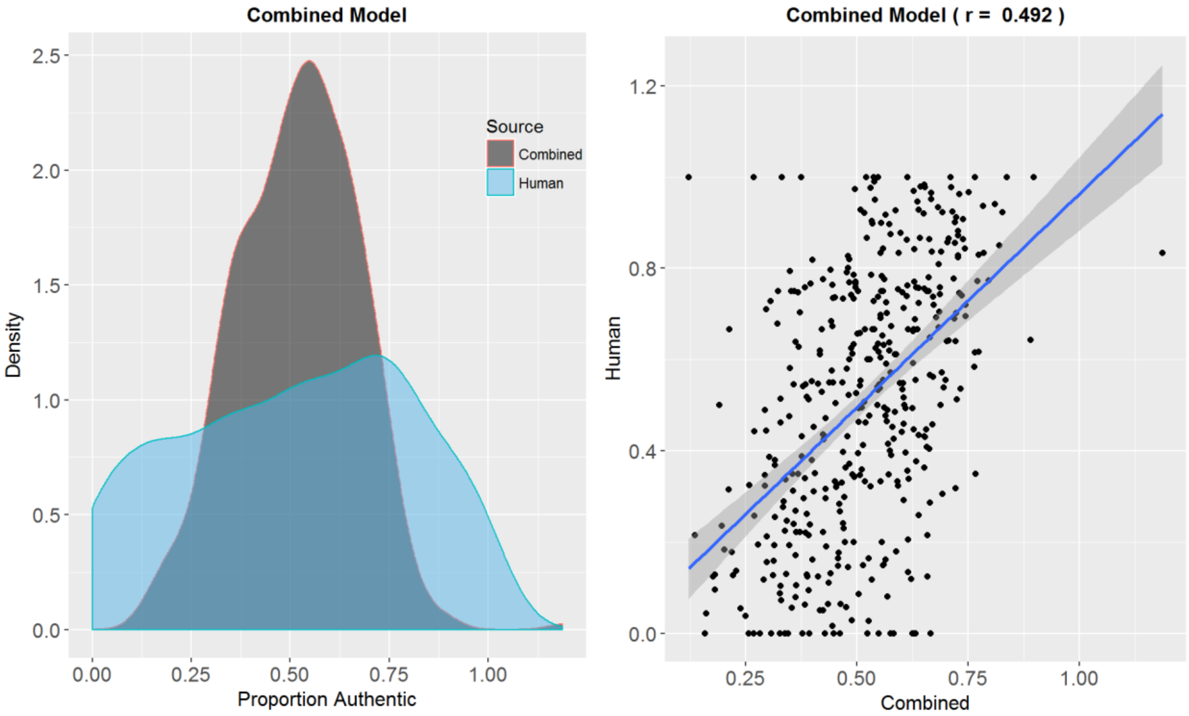


Figure 7. Density plot and scatter plot showing the resulting predictions from combining both the open and closed vocabulary models on the Partnership dataset compared to human codes.

3.4 Feature Analysis

We investigated the features (words and phrases) from the best open vocabulary model in the form of word clouds[1] scaled using correlations of individual features with authenticity rather than by absolute frequency in the corpus. Figure 8 shows words that positively correlate with authenticity for the Class 5 dataset. The words "Question," "Maybe," and "Ok" correlated most strongly with authenticity (correlation values of .254, .229, and .219 respectively). These words are used to ask questions, indicate uncertainty, or to accept another's response. This might suggest the teacher is setting the stage for open dialogue, which is precisely what authentic questioning signals.

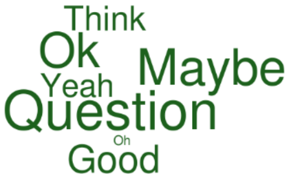


Figure 8. Words that are positively correlated with authenticity in the Class 5 dataset.

Alternatively, the words "Need," "Work," and "Doing" were most negatively correlated with authenticity (correlation values of -.383, -.330, and -.302 respectively) – see Figure 9 for the full word cloud. These words might be more likely to occur during non-dialogic activities, such as lecture or individual work.

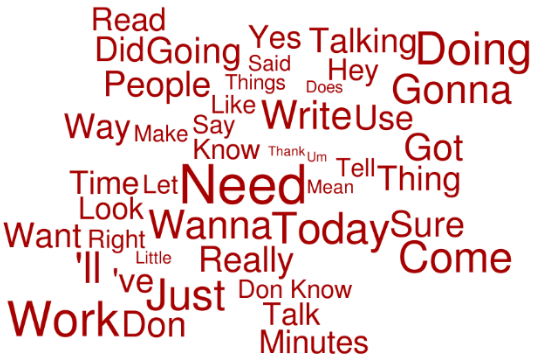


Figure 9. Words and phrases that are negatively correlated with authenticity for the Class 5 dataset.

For the Partnership dataset, only "Like," "Think," and "Say" were positively correlated with authenticity (correlation values of .177, .158, and .055 respectively). It is plausible that these terms accompany more open-ended authentic questions (e.g., "Why do you *like* the last story?" or "What do you *think* about that?" or "Why did you *say* that?") compared to their non-authentic counterparts that solicit specific responses (e.g., "What do we *know* about the beginning?" – these are all hypothetical examples).

There were also only three words that negatively correlated with authenticity. "Does" was more strongly correlated than "Know" and "Did" (correlation values of -.246, -.062, and -.032 respectively). "Does" might be more likely to accompany information-seeking questions, such as "What *does* mandible mean?" or "How *does* Jim know he is in danger?" compared to more authentic questions. Of course, these are only speculative suggestions that need to be verified by more systematic analyses.

4. DISCUSSION

We addressed the task of automated prediction of the proportion of authentic questions in a class session from real-world classroom discourse. We compared a previous closed vocabulary approach to an open vocabulary approach, combined the two, and tested them on two datasets. In the remainder of this section, we discuss our main findings, possible applications of this work, as well as limitations and directions for future work.

4.1 Main Findings

We found that the open and closed vocabulary approaches yielded equitable performance on both datasets, but a simple combination of the two resulted in statistically better results. This suggests that knowledge of the domain, as reflected in some of the closed vocabulary features (the question specific ones), is very important, but missed patterns can be captured using the open vocabulary approach. Thus, the combined approach capitalized on the strengths while mitigating the weaknesses of each individual approach.

The fact that the result replicated across two rather different datasets increases our confidence in the findings. This is particularly important because the datasets differ in a number of substantial ways – for example, one contained ASR transcripts of entire class sessions while the other contained human transcriptions of question text; one was much more variable, larger in size, and was validated at the school-level compared to the smaller, more homogenous dataset that was validated at the teacher level.

The open vocabulary approach provided key insights into the specific words used to guide its predictions. Of particular interest was the fact that the word "think" was positively correlated with authenticity in both datasets, but the word "like" was negatively correlated with authenticity in one and positively in another. This suggests the importance of examining the broader context in which these words appear.

4.2 Applications

Like anyone, teachers need feedback to improve. But in contrast to an expert musician or athlete who receives continual feedback across the countless hours spent in practice for the occasional performance, a teacher delivers approximately 1,000 "performances" a year with almost no feedback [22, 60]. Given the pivotal role of feedback to learning [5, 14, 21, 57], the lack of immediate and objective feedback is a critical barrier that needs to be cracked if we are truly going to innovate teaching.

Accordingly, one key application of our work is in an automated teacher feedback system with the goal of improving teaching effectiveness and consequently student learning. Such a system needs to be able to detect different measures of teaching effectiveness beyond authentic questions (e.g., goal clarity, disciplinary concepts, strategy use, elaborated feedback), and the open vocabulary approach is particularly suited for this task.

Ultimately, we envision technology that will autonomously analyze teachers' behaviors as they go about their daily activities, both within and beyond the classroom. The technology would provide formative feedback (i.e., feedback aimed at improvement rather

[1] Word clouds were generated via https://worditout.com

than evaluation [57]), which the teacher can use as a form of DIY (do it yourself) professional development or share with support staff. The feedback can enable reflective practice, defined as thoughtfully considering one's own actions and experiences to refine one’s skill in a selected discipline [55]. Due to its emphasis on contextualized analysis and metacognition, reflective practice holds great promise in improving teaching effectiveness [9, 10], which should result in positive downstream influences on student achievement given the robust relationship between the two [12, 17, 29, 34, 51, 52, 65].

Such a technology can also be used to streamline research into teaching effectiveness, which currently relies on cumbersome human observation (see the introduction). Going beyond question authenticity, at a broader level, such a technology could be used to advance basic research on student-teacher discourse, essentially opening up the methods of “big data” science to real-world classrooms.

4.3 Limitations & Future Work

One limitation of this study is the amount and variety of classroom transcriptions with corresponding authenticity labels. The Class 5 dataset was collected in a very limited geographical location. The Partnership dataset, although much more variable in terms of the sample, only included transcriptions of questions rather than transcriptions of all teacher utterances.

Our models also detect authenticity at the level of an entire class session, rather than at the individual utterance level. Finer grain size is needed to provide actionable feedback to teachers, at least with respect to the vision articulated above. We also did not correlate our results with more objective measures, particularly achievement growth, due to a lack of available data.

In addition to addressing the aforementioned limitations, future work should include using the open vocabulary approach to predict measures beyond authenticity. We are taking a step in this direction by re-coding current CLASS 5 audio as well as collecting new audio files and coding them for the following broader dimensions of discourse linked, or hypothesized to be linked, to student achievement growth: goal clarity, disciplinary concepts, and strategy use for teacher-led discourse, and challenge, connection, and elaborated feedback for transactional discourse.

We are also streamlining the data collection process, essentially providing usable tools for teachers to collect their own data, and have collected over 65 hours of audio (in about two months) using this approach. When coupled with existing data from CLASS 5, we estimate that the combined datasets will be sufficiently large to experiment with deep natural language processing methods, such as long short-term recurrent neural networks [31] and hierarchical attention networks [64].

4.4 Concluding Remarks

We applied an open vocabulary approach to the task of predicting authentic questions in classroom discourse and compared it to a previous closed vocabulary approach applied to the same problem. We found that the two approaches yielded equivalent performance, but a combination led to higher accuracies than either method alone. We achieved a correlation of close to 0.70 on real-world audio, which suggests that fully-automated methods might complement or even replace humans on the difficult task of determining the level of dialogism in classroom discourse.

5. ACKNOWLEDGMENTS

We acknowledge the CLASS 5 team for their contributions to the study. This research was supported by the National Science Foundation (NSF IIS 1735785) and Institute of Education Sciences (IES R305A130030). Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author and do not represent the views of the funding agencies.

6. REFERENCES

[1] Aichroth, P., Björklund, J., Stegmaier, F., Kurz, T., and Miller, G. 2015. *State of the art in cross-media analysis, metadata publishing, querying and recommendations*. Technical Report. Media in Context (MICO).

[2] American Institutes for Research. 2013. *Databases on state teacher and principal evaluation policies*. Retrieved from http://resource.tqsource.org/stateevaldb/Compare50States.aspx.

[3] Anscombe, F. J. 1948. The transformation of poisson, binomial and negative binomial data. *Biometrika*. 35, 3/4 (Dec. 1948), 246-254. DOI= https://doi.org/10.1093/biomet/35.3-4.246.

[4] Archer, J., Cantrell, S., Holtzman, S. L., Joe, J. N., Tocci, C. M., and Wood, J. 2016. *Better Feedback for Better Teaching: A Practical Guide to Improving Classroom Observations*. Jossey-Bass, San Francisco, CA.

[5] Azevedo, R. and Bernard, R. M. 1995. A meta-analysis of the effects of feedback in computer-based instruction. *Journal of Educational Computing Research*. 13, 2 (Sep. 1995), 111-127. DOI= https://doi.org/10.2190/9LMD-3U28-3A0G-FTQT.

[6] Blanchard, N. et al. 2016. Identifying Teacher Questions Using Automatic Speech Recognition in Classrooms. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue* (Los Angeles, CA, USA, September 13 - 15, 2016). 191-201.

[7] Blei, D. M. D., Ng, A. Y., and Jordan, M. I. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*. 3, 1 (Jan. 2003), 993-1022. DOI= https://doi.org/10.1162/jmlr.2003.3.4-5.993.

[8] Boakye, K., Favre, B., and Hakkani-Tür, D. 2009. Any questions? Automatic question detection in meetings. In *Proceedings of the 11th Biannual IEEE Workshop on Automatic Speech Recognition and Understanding (*Merano, Italy, December 13 - 17, 2009). ASRU '09. IEEE, Piscataway, NJ, 485-489.

[9] Camburn, E. M. 2010. Embedded Teacher Learning Opportunities as a Site for Reflective Practice: An Exploratory Study. *American Journal of Education*. 116, 4 (Jun. 2010), 463-489. DOI= https://doi.org/10.1086/653624.

[10] Camburn, E. M. and Han, S. W. 2015. Infrastructure for teacher reflection and instructional change: An exploratory study. *Journal of Educational Change*. 16, 4 (Nov. 2015), 511-533. DOI= https://doi.org/10.1007/s10833-015-9252-6.

[11] Chawla, N. V. 2005. Data Mining for Imbalanced Datasets: An Overview. In *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Springer, Boston, MA, 875-886. DOI= https://doi.org/10.1007/978-0-387-09823-4_45.

[12] Chetty, R., Friedman, J. N., and Rockoff, J. E. 2014.

Measuring the Impacts of Teachers I: Evaluating bias in teacher value-added estimates. *American Economic Review*. 104, 9 (Sep. 2014), 2593-2632. DOI= https://doi.org/10.3386/w19423.

[13] Church, K. W. and Hanks, P. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*. 16, 1 (Mar. 1990), 22-29.

[14] D'Mello, S. K., Lehman, B., and Person, N. K. 2010. Expert tutors feedback is immediate, direct, and discriminating. In *Proceedings of the 23rd Florida Artificial Intelligence Research Society Conference* (Daytona Beach, Florida, USA, May 19 - 21, 2010). AAAI, Palo Alto, CA, 504-509.

[15] D'Mello, S. K. et al. 2015. Multimodal Capture of Teacher-Student Interactions for Automated Dialogic Analysis in Live Classrooms. In *Proceedings of the 2015 ACM International Conference on Multimodal Interaction* (London, United Kingdom, January 19 - 20, 2015). ICMI '15. ACM, New York, NY, 557-566. DOI= https://doi.org/10.1145/2818346.2830602.

[16] Danielson, C. 2007. *Enhancing Professional Practice: A Framework for Teaching*. Association for Supervision and Curriculum Development, Alexandria, VA.

[17] Darling-Hammond, L. 2000. Teacher Quality and Student Achievement. *Education policy analysis archives*. 8, 1 (Jan. 2000), 1-44. DOI= https://doi.org/10.14507/epaa.v8n1.2000.

[18] Donnelly, P. J. et al. 2016. Automatic Teacher Modeling from Live Classroom Audio. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization* (Halifax, Canada, July 13 - 16, 2016). UMAP '16. ACM, New York, NY, 45-53. DOI= https://doi.org/10.1145/2930238.2930250

[19] Donnelly, P. J. et al. 2016. Multi-sensor modeling of teacher instructional segments in live classrooms. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction* (Tokyo, Japan, November 12 - 16, 2016). ICMI '16. ACM, New York, NY, 177-184. DOI= https://doi.org/10.1145/2993148.2993158.

[20] Donnelly, P. J., Blanchard, N., Olney, A. M., Kelly, S., Nystrand, M., and D'Mello, S. K. 2017. Words Matter: Automatic Detection of Teacher Questions in Live Classroom Discourse using Linguistics, Acoustics, and Context. In *Proceedings of the Seventh International Learning Analytics and Knowledge Conference* (Vancouver, BC, Canada, March 13 - 17, 2017). LAK '17. ACM, New York, NY, 218-227. DOI= https://doi.org/10.1145/3027385.3027417.

[21] Ericsson, K. A., Krampe, R. T., and Tesch-Römer, C. 1993. The role of deliberate practice in the acquisition of expert performance. *Psychological Review*. 100, 3 (Jul. 1993), 363. DOI= https://doi.org/10.1037/0033-295X.100.3.363.

[22] Fadde, P. J. and Klein, G. A. 2010. Deliberate performance: Accelerating expertise in natural settings. *Performance Improvement*. 49, 9 (Oct. 2010), 5-14. DOI= https://doi.org/10.1002/pfi.20175.

[23] Frank, E., Wang, Y., Inglis, S., Holmes, G., and Witten, I. H. 1998. Using model trees for classification. *Machine Learning*. 32, 1 (Jul. 1998), 63-76.

[24] Gamoran, A. and Kelly, S. 2003. Tracking, instruction, and unequal literacy in secondary school English. In *Stability and change in American education: Structure, process, and outcomes*, M. T. Hallinan et al., Eds. Eliot Werner Publications Incorporated, Clinton Corners, NY, 109-126.

[25] Gamoran, A. and Nystrand, M. 1992. Taking students seriously. In *Student Engagement and Achievement in American Schools*, F. M. Newman, Ed. Teachers College Press, New York, NY, 40-61.

[26] Goe, L., Biggers, K., and Croft, A. 2012. *Linking Teacher Evaluation to Professional Development: Focusing on Improving Teaching and Learning.* Research & Policy Brief. National Comprehensive Center for Teacher Quality.

[27] Grossman, P., Greenberg, S., Hammerness, K., Cohen, J., Alston, C., and Brown, M. 2009. Development of the protocol for language arts teaching observation (PLATO). In *annual meeting of the American Educational Research Association* (San Diego, California, USA, 2009).

[28] Hamilton, L. 2012. Measuring teaching quality using student achievement tests: Lessons from educators' response to No Child Left Behind. In *Assessing teacher quality: Understanding teacher effects on instruction and achievement*, S. Kelly, Ed. Teachers College Press, New York, NY, 49-76.

[29] Hanushek, E. A. and Rivkin, S. G. 2006. Teacher quality. In *Handbook of the Economics of Education*, E. A. Hanushek and F. Welsh, Eds. North-Holland, Amsterdam, The Netherlands, 1051-1078.

[30] Harris, D. N., Ingle, W. K., and Rutledge, S. A. 2014. How Teacher Evaluation Methods Matter for Accountability: A Comparative Analysis of Teacher Effectiveness Ratings by Principals and Teacher Value-Added Measures. *American Educational Research Journal*. 51, 1 (Feb. 2014), 73-112. DOI= https://doi.org/10.3102/0002831213517130.

[31] Hochreiter, S. and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*. 9, 8 (Nov. 1997), 1735-1780. DOI= https://doi.org/10.1162/neco.1997.9.8.1735.

[32] Jennings, J. L. and Corcoran, S. P. 2012. Beyond high-stakes tests: Teacher effects on other educational outcomes. In *Assessing teacher quality: Understanding teacher effects on instruction and achievement*, S. Kelly, Ed. Teachers College Press, New York, NY, 77-95.

[33] Juzwik, M. M., Borsheim-Black, C., Caughlan, S., and Heintz, A. 2013. *Inspiring dialogue: Talking to learn in the English classroom*. Teachers College Press, New York, NY.

[34] Kane, T., Kerr, K., and Pianta, R. 2014. *Designing teacher evaluation systems: New guidance from the measures of effective teaching project*. Jossey-Bass, San Francisco, CA.

[35] Kane, T. J., McCaffrey, D. F., Miller, T. and Staiger, D. O. 2013. *Have We Identified Effective Teachers? Validating Measures of Effective Teaching Using Random Assignment*. Bill & Melinda Gates Foundation, Seattle, WA.

[36] Kane, T. J. and Staiger, D. O. 2012. *Gathering Feedback for Teaching: Combining High-Quality Observations with Student Surveys and Achievement Gains*. Bill & Melinda Gates Foundation, Seattle, WA.

[37] Kelly, S., Olney, A. M., Donnelly, P. J., Nystrand, M., and D'Mello, S. K. Automatically Measuring Question Authenticity in Real-World Classrooms. *In Review*.

[38] Lin, D. 1998. Extracting collocations from text corpora. In

First Workshop on Computational Terminology (Montreal, Canada, August 15, 1998). 57-63.

[39] McKeown, M. G. and Beck, I. L. 2015. Effective classroom talk is reading comprehension instruction. In *Socializing intelligence through academic talk and dialogue*, L. B. Resnik et al., Eds. American Educational Research Association, Washington, D.C., 51-62.

[40] Mehan, H. 1979. *Learning Lessons: Social Organization in the Classroom*. Harvard University Press, Cambridge, MA.

[41] Nystrand, M. 1988. *CLASS (Classroom language assessment system) 2.0: A Windows laptop computer system for the in-class analysis of classroom discourse*. Wisconsin Center for Education Research, Madison, WI.

[42] Nystrand, M. and Gamoran, A. 1997. The big picture: Language and learning in hundreds of English lessons. In *Opening dialogue: Understanding the dynamics of language and learning in the English classroom*. M. Nystrand, Ed. Teachers College Press, New York, NY, 30-74.

[43] Olney, A. M., Samei, B., Donnelly, P. J., and D'Mello, S. K. 2017. Assessing the Dialogic Properties of Classroom Discourse: Proportion Models for Imbalanced Classes. In *Proceedings of the 10th International Conference on Educational Data Mining* (Wuhan, China, June 25 - 28, 2017). EDM '17. 162-167.

[44] Olney, A. M., Louwerse, M., Matthews, E., Marineau, J., Hite-Mitchell, H., and Graesser, A. 2003. Utterance Classification in AutoTutor. In *Proceedings of the Human Language Technology - North American Chapter of the Association for Computational Linguistics 03 Workshop on Building Education Applications Using Natural Language Processing* (Philadelphia, PA, May 31, 2003). Association for Computational Linguistics, Stroudsburg, PA, 1-8.

[45] Orosanu, L. and Jouvet, D. 2015. Detection of sentence modality on French automatic speech-to-text transcriptions. In *Proceedings of International Conference on Natural Language and Speech Processing* (Algiers, Algeria, October 18 - 19, 2015). IEEE.

[46] Park, G. et al. 2015. Automatic personality assessment through social media language. *Journal of Personality and Social Psychology*. 108, 6 (Jun. 2015), 934-952. DOI= https://doi.org/10.1037/pspp0000020.

[47] Pedregosa, F. et al. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 12, 1, (Oct. 2011), 2825-2830.

[48] Pennebaker, J. W., Mehl, M. R., and Niederhoffer, K. G. 2003. Psychological Aspects of Natural Language Use: Our Words, Our Selves. *Annual Review of Psychology*. 54, 1 (Feb. 2003), 547-577. DOI= https://doi.org/10.1146/annurev.psych.54.101601.145041.

[49] Pennebaker, J. W., Chung, C. K., Ireland, M., Gonzales, A., and Booth, R. J. 2007. *The Development and Psychometric Properties of LIWC2007*. LIWC.net, Austin, TX. DOI= https://doi.org/10.1068/d010163.

[50] Resnick, L., Michaels, S., and O'Connor, C. 2010. How (well structured) talk builds the mind. In *Innovations in educational psychology, Perspectives on learning, teaching, and human development*, D. Preiss and R. J. Sternberg, Eds. Springer, Boston, MA, 163-194.

[51] Rivkin, S. G., Hanushek, E. A., and Kain, J. F. 2005. Teachers, schools and academic achievement. *Econometrica*. 73, 2 (Mar. 2005), 417-458.

[52] Rockoff, J. E. 2004. The Impact of Individual Teachers on Student Achievement: Evidence from Panel Data. *The American Economic Review*. 94, 2 (May. 2004), 247-252. DOI= https://doi.org/10.1257/0002828041302244.

[53] Samei, B. et al. 2014. Domain Independent Assessment of Dialogic Properties of Classroom Discourse. In *Proceedings of the 7th International Conference on Educational Data Mining* (London, United Kingdom, July 04 - 07, 2014). EDM '14. 233-236.

[54] Samei, B. et al. 2015. Modeling Classroom Discourse: Do Models that Predict Dialogic Instruction Properties Generalize across Populations? In *Proceedings of the 8th International Conference on Educational Data Mining* (Madrid, Spain, June 26 - 29, 2015). EDM '15. 444-447.

[55] Schon, D. A. 1987. *Educating the Reflective Practitioner: Toward a New Design for Teaching and Learning in the Professions*. Jossey-Bass, San Francisco, CA.

[56] Schwartz, H. A. et al. 2013. Personality, Gender, and Age in the Language of Social Media: The Open-Vocabulary Approach. *PLoS ONE*. 8, 9 (Sep. 2013), e73791. DOI= https://doi.org/10.1371/journal.pone.0073791.

[57] Shute, V. J. 2008. Focus on Formative Feedback. *Review of Educational Research*. 78, 1 (Mar. 2008), 153-189. DOI= https://doi.org/10.3102/0034654307313795.

[58] Stein, M. K. and Matsumura, L. C. 2009. Measuring instruction for teacher learning. In *Measurement issues and assessment for teacher quality*, D.H. Gitomer, Ed. Sage Publications, Los Angeles, CA, 179-205.

[59] Steyvers, M. and Griffiths, T. 2007. Probabilistic Topic Models. In *Handbook of latent semantic analysis*, T. K. Landauer et al., Eds. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, 424-440.

[60] Stigler, J. and Miller, K. 2006. Expertise and Expert Performance in Teaching. In *The Cambridge Handbook of Expertise and Expert Performance*, K. A. Ericsson et al, Eds. Cambridge University Press, Cambridge, United Kingdom, 431-452.

[61] Stolcke, A. et al. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*. 26, 3 (Sep. 2000), 339-373.

[62] Wang, Z., Miller, K., and Cortina, K. 2013. Using the LENA in teacher training: Promoting student involvement through automated feedback. *Unterrichtswissenschaft*. 4, 4 (Nov. 2013) 290-305.

[63] Wilkinson, I. A. G., Soter, A. O., and Murphy, P. K. 2010. Developing a model of Quality Talk about literary text. In *Bringing reading research to life*, M. G. McKeown and L. Kucan, Eds. Guilford, New York, NY, 142-169.

[64] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (San Diego, California, June 12 - 17, 2016). Association for Computational Linguistics, Stroudsburg, PA, 1480-1489.

[65] Yoon, K. S., Duncan, T., Lee, S. W. Y., Scarloss, B., and

Shapley, K. L. 2007. *Reviewing the evidence on how teacher professional development affects student achievement*. REL 2007-No. 033. Regional Educational Laboratory Southwest (NJ1).

[66] Zou, G. Y. 2007. Toward Using Confidence Intervals to Compare Correlations. *Psychological Methods*. 12, 4 (Dec. 2007), 399-413. DOI= https://doi.org/10.1037/1082-989X.12.4.399.

Impact of Corpus Size and Dimensionality of LSA Spaces from Wikipedia Articles on AutoTutor Answer Evaluation

Zhiqiang Cai
The University of Memphis
365 Innovation Drive, Suite 410
Memphis, TN 38152-3115, USA
zcai@memphis.edu

Arthur C. Graesser
The University of Memphis
365 Innovation Drive, Suite 403
Memphis, TN 38152-3115, USA
graesser@memphis.edu

Leah C. Windsor
The University of Memphis
365 Innovation Drive, Suite 403
Memphis, TN 38152-3115, USA
lcwells@memphis.edu

Qinyu Cheng
The University of Memphis
365 Innovation Drive, Suite 410
Memphis, TN 38152-3115, USA
qcheng@memphis.edu

David W. Shaffer
University of Wisconsin-Madison
Aalborg University-Copenhagen
1025 West Johnson St
Madison, WI 53706-1706, USA
dws@education.wisc.edu

Xiangen Hu
University of Memphis
Central China Normal University
365 Innovation Drive, Suite 403
Memphis, TN 38152-3115, USA
xhu@memphis.edu

ABSTRACT

Latent Semantic Analysis (LSA) plays an important role in analyzing text data from education settings. LSA represents meaning of words and sets of words by vectors from a *k*-dimensional space generated from a selected corpus. While the impact of the value of *k* has been investigated by many researchers, the impact of the selection of documents and the size of the corpus has never been systematically investigated. This paper tackles this problem based on the performance of LSA in evaluating learners' answers to AutoTutor, a conversational intelligent tutoring system. We report the impact of document sources (Wikipedia vs TASA), selection algorithms (keyword based vs random), corpus size (from 2000 to 30000 documents) and number of dimensions (from 2 to 1000). Two AutoTutor tasks are used to evaluate the performance of different LSA spaces: a phrase level answer assessment (responses to focal prompt questions) and a sentence level answer assessment (responses to hints). We show that a sufficiently large (e.g., 20,000 to 30,000 documents) randomly selected Wikipedia corpus with high enough dimensions (about 300) could provide a reasonably good space. A specifically selected domain corpus could have significantly better performance with a relatively smaller corpus size (about 8000 documents) and much lower dimensionality (around 17). The widely used TASA corpus (37,651 documents scientifically sampled) performs equally well as a randomly selected large Wikipedia corpus (20,000 to 30,000) with a sufficiently high dimensionality (e.g., *k*>=300).

Keywords

AutoTutor, LSA, TASA, Wikipedia, corpus size, dimensionality

1. INTRODUCTION

1.1 Latent Semantic Analysis in Education Data Mining

Text mining is one of the most important tasks in education data mining [21]. Education text data could be textual learning content presented to learners, essays from learners, solutions to problems, answers to questions, conversations between collaborators, and so on. Researches have shown that analyzing such text data is crucial for improving education quality and reducing education cost. For example, Graesser et al. [9] reported that scaling texts to match the reading level and reading style of learners could facilitate the learning process. Foltz et al. [19] showed that automatic essay grading could greatly reduce teachers' workload. Wiemer-Hasting et al. [24] and Graesser et al. [10] showed that automatic answer evaluation makes it possible for intelligent environments to give immediate feedback to learners' text inputs. LSA (latent semantic analysis) plays an important role in all these text analysis tasks.

LSA is a method that extracts the meaning of words from a large body of texts (corpus) [15]. The mathematics behind LSA is surprisingly simple. The extraction process is just counting the number of occurrences of each word in each document, resulting in a word-document matrix, with rows representing words and columns representing documents. Thus, each row of the matrix is actually a vector representation of a word in a high dimensional space (the number of dimensions equals the number of documents). The raw occurrence counts are usually transformed by certain weighting method, such as TFIDF or Log-Entropy (see e.g., [14, 16]). After the transformation, a matrix entry has a higher value if the corresponding word is unevenly distributed in the corpus and frequent in the document corresponding to the column the word entry is in. A dimension reduction technique, namely, singular value decomposition, is applied to the weighted matrix to produce vector representations for words (as well as documents) with lower dimensionality. Weighted sum of word vectors is often used to form vector representations of phrases, sentences, paragraphs and documents. Different weight algorithms and their effects can be found in McNamara et al. [17]. More details on LSA vector space generation can be found in Landauer et al. [15].

With vector representations, the similarity of the meaning of two texts can be computed as the cosine between two vectors. This similarity measure has been widely used in many applications. For example, Coh-Metrix (cohmetrix.com) measures text cohesion by computing the average LSA cosine between sentence vectors and paragraph vectors [11]. AutoTutor (autotutor.org) evaluates learners' text inputs by computing the cosine between the input text vector and the ideal answer vector [4]. The Intelligent Essay Assessor [19] uses LSA cosine between vectors of target essay and pre-scored essays as one of the most important predictor in automatic essay scoring.

The number of dimensions of LSA vector spaces, usually denoted by *k*, has been investigated by many researchers. The most influential study is probably the one published by Landauer and Dumais in 1997 [14]. They generated an LSA space from 30,473 encyclopedia articles and then applied the vectors in a TOEFL (Test of English as a Foreign Language) word comparison task. They found that the value of *k* had large impact on the LSA performance and the best choice was about 300. This value, *k*=300, has been used as a magic number in many later applications. However, researchers also reported a large range of optimal values of *k* (from 6 to over 1000), depending on the corpus used for generating the LSA space and the specific task the LSA was applied to. A long list of studies can be found in Bradford (2008) [2].

In addition to dimensionality, the size and the content of the corpus used for LSA space generation also influences the performance of LSA. Researchers reported the use of different corpora, such as Touchstone Applied Science Association (TASA) corpus (http://lsa.colorado.edu/spaces.html), the Corpus of Contemporary American English (COCA) [7], Encyclopedia, and so on. The size of reported corpora varied from hundreds to hundreds of thousands of documents. Some studies reported the optimal values of *k* for different corpora with very different sizes. For example, Kontostathis (2007) [13] reported a study on 7 corpora with sizes varying from 1033 to 348,566 documents. While the optimal value of *k* for each corpus was reported, no corpus size effect was considered. A recent study reported by Crossley et al. (2017) [6] showed significant performance differences of two corpora with different sizes (44K vs 55K documents) and the same *k* =300, assuming 300 is the optimal value. However, they did not consider the real optimization of the value *k* for each corpus. Furthermore, the two corpora, TASA and COCA, contain different kinds of articles. It is not clear whether it was the corpus size or any other document features (e.g. genre, readability, concreteness, cohesion, formality, etc.) that caused the performance difference.

The size of data that needs to be stored for the word vectors of an LSA space is proportional to the product of number of words and the value of *k*. When performance is guaranteed, a small corpus with a small *k* would save both storage and computation cost in using LSA. Searching for an optimal corpus size to generate an LSA space is a task that needs to be accomplished together with the optimization of *k*. That is, it is an optimization problem involving two parameters, which unfortunately is more difficult and computationally more expensive. An added difficulty is to control other document features when the corpus size varies. One way to control the document features is to systematically sample articles from a particular single source to form corpora of different sizes. In this paper, we will focus on sampling corpora from Wikipedia.

1.2 Latent Semantic Analysis for AutoTutor Answer Evaluation

AutoTutor is a conversational intelligent tutoring system. Since the late 1990s, many AutoTutor systems have been developed, targeting different domains, including computer literacy, physics, critical thinking, and electronics. AutoTutor has become an ideal platform for collecting text data from learners and a testbed for natural language processing technologies.

AutoTutor helps learners learn by holding a conversation between computer agents (one or more) and human learners, targeting the solution to specific problems. AutoTutor usually starts a conversation with a deep "main question", to which the ideal answer is often about a paragraph in length. The goal of the conversation is to help a learner construct an answer semantically equivalent to the prepared ideal answer. To learners who cannot meet the goal at the first try, AutoTutor asks follow-up questions that target missing information. There are two types of such AutoTutor follow-up questions. One is called "hint", to which the answer is about a sentence long. Another is called "prompt", to which the answer is usually a word or a phrase. An AutoTutor conversation ends either when the goal is met or the prepared questions are exhausted. The following is an excerpt of an AutoTutor conversation in a Newtonian physics system.

AutoTutor [main question]: *Suppose a boy is in a free-falling elevator and he holds his keys motionless right in front of his face and then lets go. What will happen to the keys? Explain why.*
Student [first try]: *The keys will move up since the elevator is falling and the keys are in the elevator, the keys will go in the opposite direction of the free falling elevator.*
AutoTutor [feedback]: *Umm, that is an interesting answer!*
AutoTutor [hint]: *What can you say about the vertical velocity of the keys?*
Student [response to hint]: *The vertical velocity will be at a constant acceleration of 9.8.*
AutoTutor [feedback]: *OK. See if you can get this.*
AutoTutor [prompt]: *The boy and his keys have the same initial vertical what?*
Student [response to prompt]: *Velocity.*

In each conversation turn, AutoTutor evaluates the learner's input and makes decisions on the selection of feedback type and the next move (asking a new question or ending the conversation). With the help of regular expressions, LSA plays an important role in matching the meaning of the learner's input with prepared answers. The mathematical AutoTutor assessment models, normally a combination of LSA and regular expressions, can be found in Cai et al. (2011) [5].

When an AutoTutor application is developed, an LSA space generated from a domain specific corpus is often needed, because the meaning of words may differ from domain to domain. For example, the meaning of the word "force", according to Oxford Dictionary (https://en.oxforddictionaries.com) could be defined as "strength or energy as an attribute of physical action or movement" or "an organized body of military personnel or police". When an LSA space is generated, the meaning of a word can usually be observed by the "nearest neighbors", namely, the words with highest LSA cosine with the target word. Table 1 shows 5 nearest neighbors of the word "force" from three different LSA spaces: a Wikipedia space targeting Newtonian physics articles (4000 documents, 17 dimensions), a randomly sampled Wikipedia space (4000 articles, 17 dimensions) and the TASA corpus (37651 articles, 300 dimensions). It looks obvious that the meaning of "force" in the targeted corpus and TASA is more of the sense in Newtonian physics, while in the random Wikipedia space, the meaning is more of the sense in military.

Table 1. Nearest neighbors of "force" in different spaces

Corpus	Docs	Dim	Nearest Neighbors
Targeted	4000	17	exert, act, pull, experience, push
Random	4000	17	belligerent, offensive, gun, command, patrol
TASA	37651	300	unbalanced, exert, centripetal, turntable, Newton

It has long been believed that the performance of LSA depends on the selection of corpus. Cai et al [5] showed that, with a well selected corpus, LSA could be used together with regular expressions to build a model that evaluates learners' responses in AutoTutor almost as good as human. However, it has never been reported about the combined impact of the article selection, the size of the corpus and the optimization of *k* for the LSA component.

1.3 Wikipedia as Document Source for Corpus Sampling

In order to investigate the impact of document selection and corpus size, we need a reliable document source that contains enough many articles for different domains. Wikipedia (Wikipedia.org) is an ideal source for this. By the end of 2017, the English Wikipedia had about 5.6 million content articles, containing almost everything. New articles are still being added.

There are reports on LSA spaces generated from Wikipedia. For example, Ștefănescu et al. (2014) [22] compared the performance of Wikipedia spaces with TASA spaces on a word similarity task. However, they did not consider the "domain" specificity and the impact of the corpus size. They took all documents in Wikipedia as a whole for LSA space generation, taking into account of different filtering strategies, resulting in huge spaces.

1.4 Rational and Research Questions

Researchers have believed that the corpus used to generate an LSA space should align with the targeted domain. Gotoh et al. (1997) [8] showed a typical way of constructing a domain specific corpus: finding articles labeled in a category, such as "natural science", "world affairs", "arts", and so on. The targeted domain is then represented as a mixture of such categories. People are often convinced that domain specific spaces are needed from seeing "nearest neighbors" that show different meaning representations (see Table 1). However, several questions remain unanswered. For a given task, is it really necessary to generate a domain specific space? In other words, does a domain specific space perform significantly better than a generic space? A related question immediately emerges: how do we measure the "domain specificity"? How do we know the degree to which a corpus is targeting a given domain? Furthermore, what do we mean by a "domain"? How should a domain be defined or specified? There are also practical application questions related to this. For example, would a domain specific space save storage and computation costs with better or equivalent accuracy in performing a given task? Answers to these questions are important. If we know a generic space (e.g. TASA) can work as well as a domain specific space, we will not need to spend time and resources to generate new spaces. Domain spaces are needed only if they perform significantly better or can save storage and computation time without sacrificing performance.

2. METHOD

2.1 AutoTutor Data

We compared the performance of LSA spaces on evaluating learners' responses to a Newtonian physics AutoTutor. The data contained responses of college students to 10 problems about Newtonian physics. Table 2 shows the number of hints and prompts and the number of responses in each of the 10 problems. There were 114 hints and 133 prompts in total. This resulted in 4941 hint responses and 2643 prompt responses. On average, there were about 43 responses per hint and 20 responses per prompt. The reason why there were more hint responses was that AutoTutor conversations started with a hint, followed by a prompt, then another hint followed by another prompt, and so on. An AutoTutor conversation ended when a learner's responses covered all aspects of the ideal answer. Thus, if the conversation ended after a prompt, the number of hint responses and prompt responses in that conversation would be the same. However, if the conversation ended after a hint, the number of hint responses would be one more than the number of prompt responses in that conversation. The ratio of prompt responses to hint responses depends on the number of "hint-prompt" cycles that occurred in the conversation. The fact that there were more than twice of hint responses than prompt responses indicates that many conversations ended after the first or second hint question.

Table 2. Number of hint and prompt responses

Problem	Hints	Hint responses	Prompts	Prompt responses
Pumpkin	16	865	12	299
Sun and earth	6	186	1	23
Free key fall	12	969	10	403
Neck injury	11	308	12	272
Clown juggling	15	540	30	481
Car collision	11	431	6	86
Packet drop	13	801	9	285
Container mass	10	454	15	403
Clay balls	11	213	25	264
Car towing	9	174	13	127
Total	**114**	**4941**	**133**	**2643**

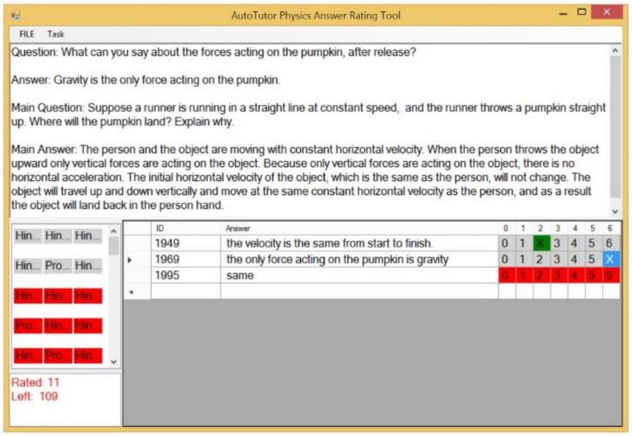
According to the design of AutoTutor, a hint question targets an answer about a sentence long and the answer to a prompt question is usually a word or a phrase. Table 3 shows that the hint responses in our data set were about 8 words on average; and the prompt responses were about 3 to 4 words on average. Penumatsa et al. (2006) [18] showed that the cosine values are length dependent. That is, longer texts tend to yield larger cosine values. Cai et al. (2016) [4] reported that LSA performed differently on hint responses and prompt responses. Their explanation was that the hint questions and prompt questions had different "uncertainties". The responses to a question with higher uncertainty would be more divergent and thus more difficult to assess. Following this, we investigated LSA performance on hint and prompt responses separately.

Table 3. Hint and prompt answer/response lengths

Answer	N	Mean	Std
Hint Ideal	112	10.64	3.59
Hint Responses	4861	8.01	8.82
Prompt Ideal	125	3.53	1.06
Prompt Responses	2603	1.64	3.06

2.2 Human Rating

The student responses were rated by two experts; one was a full professor and the other was a graduate student. Both raters had background in computer engineering and had good understanding of Newtonian physics. A rating tool was built to facilitate the rating process (see Figure 1). At the middle left panel of the tool, there is a list box for raters to choose hints and prompts. At the top panel, there is a text box that displays a hint/prompt question, together with its associated main question and their answers. The data table at the bottom right panel shows all student responses and 7 rating options. A response is scored by a click on an option. A "0" means the response is not an answer to the question at all, such as "what", "I don't know", "what do you mean", and so on. A "1" indicates a response that has no semantic similarity to the prepared ideal answer and "6" indicates a perfect answer. Red and gray colors are used to mark unrated and rated items, respectively. At the bottom left corner, there is a text box that shows the number of items already rated and the number of items that are to be rated. This tool helped the raters more easily and accurately rate the responses.



Question: What can you say about the forces acting on the pumpkin, after release?

Answer: Gravity is the only force acting on the pumpkin.

Main Question: Suppose a runner is running in a straight line at constant speed, and the runner throws a pumpkin straight up. Where will the pumpkin land? Explain why.

Main Answer: The person and the object are moving with constant horizontal velocity. When the person throws the object upward only vertical forces are acting on the object. Because only vertical forces are acting on the object, there is no horizontal acceleration. The initial horizontal velocity of the object, which is the same as the person, will not change. The object will travel up and down vertically and move at the same constant horizontal velocity as the person, and as a result the object will land back in the person hand.

ID	Answer	0	1	2	3	4	5	6
1949	the velocity is the same from start to finish.	0	1	X	3	4	5	6
1969	the only force acting on the pumpkin is gravity	0	1	2	3	4	5	X
1995	same	0	1	2	3	4	5	6

Rated: 11
Left: 109

Figure 1. Rating tool.

From the 7584 responses, 120 were randomly sampled as training corpus. After rating the training corpus, two raters discussed the rating criteria and independently rated the rest of the items. Table 4. shows the correlations between the two raters. The correlations were about 0.82, which indicate that there were some disagreements between the two raters. In other words, even for human experts, such evaluation tasks are sometimes difficult. We had thought that answers to prompts should be easier to evaluate than answers to hint. However, the correlation of two raters' ratings on prompts is only slightly higher. The Fisher transform [12] showed that the Z value of the two correlations (hints vs prompts) was 0.90 (*p*=0.369), which indicates that the difference is not significant. It should be noted that this Z value is for two independent correlations from different samples. There is another Z-test for dependent correlations, which will be used in the later part of this paper. The Z transform showed that the human rates agreed on hint and prompt responses similarly. That indicates that human raters did not experience more difficulty in evaluating hint responses than prompt responses.

Table 4. Correlations between ratings of two raters.

Question Type	N	Correlation
Hint	4861	0.820
Prompt	2603	0.827
All	7464	0.828

2.3 Sampling Corpora from Wikipedia

2.3.1 *Seeding method for sampling domain specific corpus*

Our goal was to investigate whether or not a "domain specific" corpus generates an LSA space with higher performance for our tasks. However, it is hard to quantify what a "domain specific" corpus really is. Many researchers used corpora that showed obvious domain labels. For example, MED corpus is for "Medical", CISI corpus for "Information Science" [3], COCA for "Contemporary American English" [7], and so on. However, we don't really know how specific these corpora are with respect to the labeled domain.

Our way of handling this problem starts from specifying a domain by a seed corpus – a small number of documents representing the targeted domain. The seed corpus could be the sections of a book, a small collection of articles focusing on a specific topic, or just some documents that are under analysis.

Once a seed corpus is identified, we extract keywords out of the seed corpus and assign a "keyness" value to each keyword. Thus, a "domain" is represented by the keyness assignment to the domain vocabulary. This is similar to the idea in topic modeling, where a topic is represented by probability distribution on a word list (see, for example, [1]).

The word keyness computation is then applied to compute document keyness by averaging the keyness of the words in a document. To search documents from a large document source (such as Wikipedia), we compute the keyness of each document. The documents in the source are then ranked by the document keyness. We select the high ranking documents from the source as a domain corpus. We call this process the "seeding method."

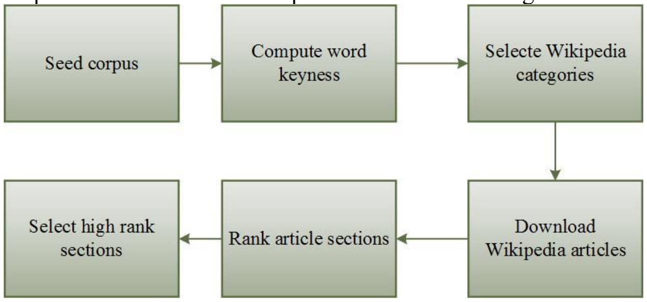

Seed corpus
Compute word keyness
Selecte Wikipedia categories
Download Wikipedia articles
Rank article sections
Select high rank sections

Figure 2. Illustration of seeding method.

Figure 2 visually illustrates the process of selecting documents from Wikipedia articles. It is difficult to directly evaluate the validity of this process. A possible way is to present a sample of documents to experts and see what proportion of documents are highly relevant to the desired domain. We do not do so in this paper. Instead, we evaluate this process by comparing the LSA performance of selected corpora with randomly sampled corpora. Our logic is simple: if domain specificity matters and the selected corpora work better than random corpora, then the seeding method is valid.

2.3.2 *Computing word keyness*

To compute the keyness of words, we considered two factors. First, a high keyness word should not be very common in general use. To quantify this, we used the log-entropy weight from a general reference corpus, TASA, as a measure of how common a word is in general use:

$$E(w) = 1 + \frac{\sum_{i=1}^{N} p_i(w) log p_i(w)}{log N}$$

where

$$p_i(w) = \frac{frequency\ of\ w\ in\ document\ i}{total\ frequency\ of\ w\ in\ the\ corpus}$$

In the above equations, N is the number of documents in TASA, which is 37,651. The log-entropy weight, $E(w)$, ranges from 0 to 1. A value close to 0 indicates that the word w is evenly distributed in TASA corpus, such as function words. A value close to 1 indicates that the word distributed unevenly in the corpus. More detailed information about entropy use can be found in LSA publications (for example, Martin et al. (1994) [16]).

Another factor we considered was that a high keyness word should be highly frequent in the seed corpus. We used the normalized logarithm of frequency to quantify this. The final word keyness with respect to the seed corpus was computed as the product of two values. One was the logarithm of the number of seed documents the word is in, divided by the logarithm of the total number of documents; and the other was the log-entropy weight of the word in TASA:

$$keyness = \frac{\log f(w)}{\log D} E(w)$$

where

$f(w)$: the number of seed documents the word w is in;

D: the total number of seed documents; and

$E(w)$: the log-entropy weight of the word w from TASA corpus.

2.3.3 *Sampling a Newtonian physics corpus from Wikipedia*

For this study, we used the 114 hint questions and the 133 prompt questions as seed corpus to compute the word keyness. This is such a small corpus that it only covered a small part of Newtonian physics. However, it provided a good starting point for us to find related categories from Wikipeida. Using the keyness equation, each word in the seed corpus was assigned a keyness. We ignored the words with keyness less than 0.01 and obtained 262 keywords. The top 10 keywords, together with their keyness values are listed below:

1) free-fall: 0.588
2) packet: 0.537
3) pumpkin: 0.526
4) acceleration: 0.496
5) velocity: 0.478
6) clown: 0.467
7) velocities: 0.449
8) horizontal: 0.427
9) keys: 0.424
10) headrest: 0.407

From the list above, we see that some words, such as “acceleration”, “velocity” are concepts of Newtonian physics. However, other words are specific to the 10 problems. To construct a corpus that has a wide coverage of Newtonian physics, we queried Wikipedia categories with these 262 keywords and obtained 154 associated categories. From these 154 categories we manually selected 16 categories that are highly related to Newtonian physics, as shown in the list below:

1) Acceleration
2) Change
3) Classical mechanics
4) Concepts in physics
5) Dynamics(mechanics)
6) Force
7) Gravitation
8) Kinematics
9) Mass
10) Mechanics
11) Motion
12) Physics
13) Systems
14) Temporal rates
15) Time
16) Velocity

In Wikipedia, each category is associated with a set of articles and a set of subcategories. For example, at the time this paper was written, the category “force” contained 67 articles (such as “force”, “friction”, “weight”, etc.) and 8 subcategories (such as “motion”, “fictitious forces”, “friction”, etc.) The above 16 selected categories served as “seed categories” of our query. We downloaded all articles from these 16 categories. Then we downloaded the articles from subcategories. Since each subcategory also contained subcategories, we could actually find a very large number of articles by following the subcategories of subcategories. In this study, we downloaded 30,000 articles.

We did not treat each article as a document of our corpus for LSA space generation. Instead, we used selected sections in the articles. Each Wikipeida article contained a definition section and many other sections. For example, the article “force” in physics contained 17 sections, such as “Development of the concept”, “Pre-Newtonian concepts”, “Newtonian mechanics”, etc. We computed the keyness of each section of each article by averaging the word keyness computed from the seed corpus. Words that did not appear in the seed corpus or with keyness less than 0.01 were ignored. Notice that, although the problem specific keywords, such as “packet”, “pumpkin”, “clown”, etc., had high keyness, since they are unlikely to appear in the articles from the selected categories, their effect in the section selection process was limited. The list below shows the number of sections of the top 10 keywords appeared in a corpus with 32,000 selected sections:

1) free-fall:73
2) packet:106
3) pumpkin:7
4) acceleration:1906
5) velocity:4424
6) clown:4
7) velocities:866
8) horizontal:1204
9) keys:124
10) headrest:1

Obviously, the Newtonian physics concepts, such as “velocity”, “acceleration”, etc., dominated the selection process. The problem specific terms, such as “headrest”, “clown” and “pumpkin” rarely appeared in the selected sections.

To avoid section length effect, we ignored any section with length < 50 words. For sections with words between 50 and 300, the section keyness was the average word keyness. For long sections with more than 300 words, the keyness was computed as the average over the first 300 words.

To compare the impact of the corpus size, we selected 5 corpora with size (number of sections) 2,000, 4,000, 8,000, 16,000 and 30,000. We name them NP2000, NP4000, …, NP30000, where "NP" stands for "Newtonian Physics". Each Newtonian physics corpus contained the highest keyness sections in the selected articles. Therefore, they were nested, namely, the sections of a smaller Newtonian physics corpus were all included in a larger Newtonian physics corpus. For example, NP8000 contained all sections of NP2000 and NP4000 (see Figure 3, left).

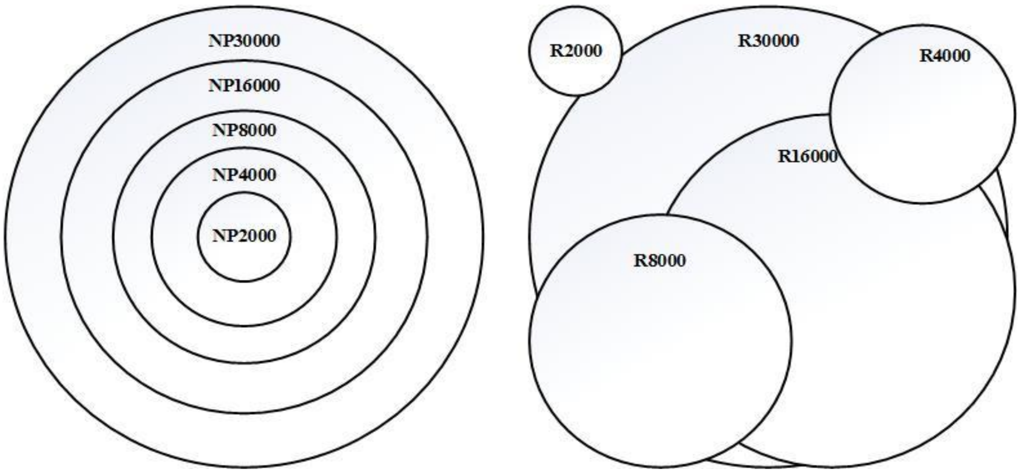


Figure 3. Illustration of nested domain corpora (left) and overlapped random corpora (right).

2.3.4 Sampling a random corpus

In order to compare the effect of the keyness-based sampling, we randomly sampled 5 corpora with same sizes as Newtonian physics corpora. The sampling process was similar to the Newtonian physics corpora sampling. The difference was that the seed keywords were 1000 words randomly sampled from TASA vocabulary. We downloaded 30,000 articles from the categories and their subcategories associated with the 1000 seed keywords. Then we randomly sampled sections from the 30,000 articles. Like NP sampling, sections with less than 50 words were ignored. The five random corpora were named, based on their sizes, as R2000, R4000, R8000, R16000 and R30000. The random corpora could be overlapped but not necessarily nested (see Figure 3, right).

2.4 LSA Spaces

A total of 11 LSA spaces were generated, 5 Newtonian physics spaces, 5 random spaces and a TASA space. The log-entropy weighting was applied to the word-document matrices. Function words and words appeared less than 3 documents in a corpus were ignored. The dimensions were all 1,000. In the rest of the paper, we will only refer to these 11 spaces. However, the similarities in each space were computed with varied dimensions. Mathematically speaking, different dimensions means different spaces. For example, NP8000 with 100 dimensions is a different space than NP8000 with 300 dimensions. However, in this paper, we refer to them as the "same space" and treat the dimension as a parameter in computing LSA similarities.

2.5 Evaluating AutoTutor Responses by LSA

For each of the above 11 spaces, the LSA semantic similarities between ideal answers and learners' responses were computed for the varying number of dimensions (k=2, 3, …, 1000). The performance of each space with each value of k was measured by the correlation between the LSA similarity and the average human rating on the responses. Cai et al [4] showed that LSA performances on hint questions and prompt questions are very different. Therefore, we considered the LSA performance on hint questions and prompt questions separately. Table 5 shows some example responses of a hint question, their LSA similarity to the ideal answer, and the human rating.

Table 5. Example of learners' responses to the hint question "How does the net force affect the car?". The ideal answer is "The net force exerted on the car results in an acceleration of two meters per second squared." LSA similarities were computed using TASA space, 300 dimensions. Human ratings are average scores of two raters.

Response	LSA	Human
Horizontally	0.16	1
it stays the same	0.16	1
it does not effect the car	0.19	1
it causes it to accelerate	0.49	2
the net force doesn't change and therefore when the mass is doubled the acceleration must be halved	0.51	5.5
The net force on the car is what causes it to accelerate	0.69	5.5
It causes an acceleration of two meters per second	0.71	5.5

The significance of performance differences were measured by Steiger Z-test [23], which is a statistic method for testing the significance of differences between two dependent correlations that share a variable in common. This is different from the independent correlation comparison that we used earlier. The Steiger Z-test compares correlation coefficients involving three variables. Assuming the three variables are A, B and C, with C as the shared common variable, the two correlations under comparison are:

- The correlation coefficient between A and C and
- The correlation coefficient between B and C.

To compare the two correlation coefficients, the correlation coefficient between A and B is also included in the computation, together with the number of data points, *N*. When the absolute value of Z is greater than 1.96, the two correlation coefficients under comparison are considered significantly different. In our study, C is the human rating whereas and A and B are two LSA similarities.

3. RESULTS

3.1 Impact of corpus size and number of dimensions for Newtonian physics spaces

Consider first the hint responses. Although the corpus sizes were very different among the five Newtonian physics spaces, the performance curves as functions of dimensions were surprising similar. They all had lowest performance at *k*=2, with correlations about 0.28. When the number of dimensions increased, the performance curves of all spaces quickly increased. The peak of about 0.425 was reached around *k*=17. The performance curves then dropped and reached a trough around *k*=128. After that, they grew up again and converged from about *k*=300 to a value about 0.40. Figure 4 shows the performance as functions of *k* for the NP spaces on hint responses from *k*=2 to *k*=1000. We used a logarithm scale on dimensions, following the method that Landauer et al. (1998) used when plotting the dimensionality effects on TOEFL tests.

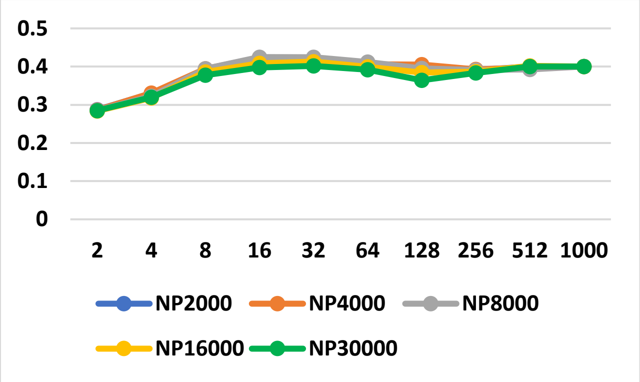


Figure 4. Performance of NP spaces on hint responses.

Although not very large, significant differences were observed among different spaces. The best performance on hint responses was NP8000 at *k*=17, with the highest correlation being 0.428. Z-test showed that, with the same *k* value, NP8000 performed significantly better than other spaces (see Table 6). The differences of correlations were from 0.01 to 0.036. This value, 0.428, was also significantly better than the performance of the same space NP8000 with *k* value less than 16 or greater than 64 (see Table 7).

Table 6. Z-test comparing performance of target spaces with fixed *k*=17 and varied corpus size on hints to the optimal target space (corpus size=8000, *k*=17) and performance (0.428). R-opt is the correlation with the optimal space. N=4861.

Space	Performance	R-opt	Z	p(2-tail)
NP2000	0.415	0.916	2.449	0.014
NP4000	0.418	0.964	2.837	0.004
NP16000	0.409	0.970	5.654	0.000
NP30000	0.392	0.934	7.607	0.000

Table 7. Z-test comparing performances of target spaces with fixed corpus size=8000 and varied *k* on hints to the optimal target space (corpus size=8000, *k*=17) and performance (0.428). R-8000 is the correlation to the optimal space. N=4861.

Dim	Performance	R-opt	Z	p(2-tail)
2	0.286	0.582	11.837	0
4	0.322	0.731	11.031	0
8	0.394	0.913	6.265	0
16	0.425	0.997	2.986	0.003
32	0.425	0.953	0.756	0.45
64	0.412	0.88	2.525	0.012
128	0.395	0.794	3.976	0
256	0.391	0.732	3.916	0
512	0.393	0.689	3.448	0
1000	0.396	0.655	3.001	0.003

The Newtonian physics spaces performed differently on prompt questions. The overall performance on prompts were higher than on hints. Also, the corpus size had a larger impact. NP8000 performed best overall. Two smaller spaces, NP2000 and NP4000, performed significantly worse. Larger spaces performed almost equally as well as NP8000. For *k*=2, the performance of smaller spaces was around 0.2, while for larger spaces, the performance was over 0.3. The performance curves for all spaces increased when the value of *k* increased. However, there was no early peak. At about *k*=24, the performance curves started to converge. The best performance for *k*=24 was again the space NP8000, which was 0.542. When the value of *k* further increased, the performance curves continuously and slowly increased. The maximum performance was at about *k*=300, which is 0.566 for larger spaces. The performance curves slowly dropped after *k*=300. At *k*=1000, the performance of small spaces was about 0.51 and the larger spaces around 0.54. Figure 5 shows the performance curves of the spaces as functions of *k* on prompt responses.

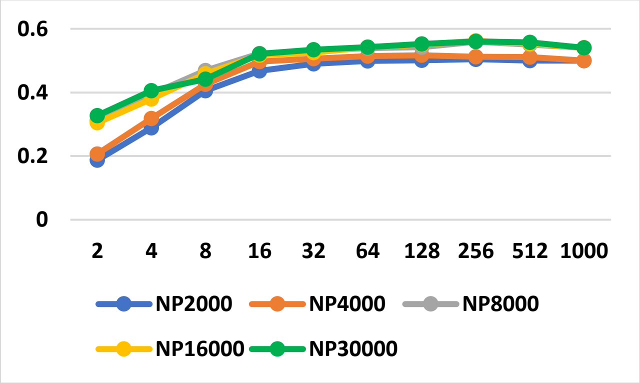


Figure 5. Performance of NP spaces on prompt responses

3.2 Comparing with Random Wikipedia Spaces and TASA Space

For small values of *k* (<32), Newtonian physics spaces performed much better (about 0.1 higher) than random spaces and TASA space on both hint responses and prompt responses. TASA space was worse than Newtonian physics spaces but better than random spaces. However, the performance curves of all large spaces converged to almost the same after about 300 dimensions. Figure 6 shows the performance of NP8000, R8000 and TASA on hint responses. Unlike Newtonian physics spaces, random spaces and TASA space did not have a peak performance. Instead, their performance curves continuously and slowly grew and converged.

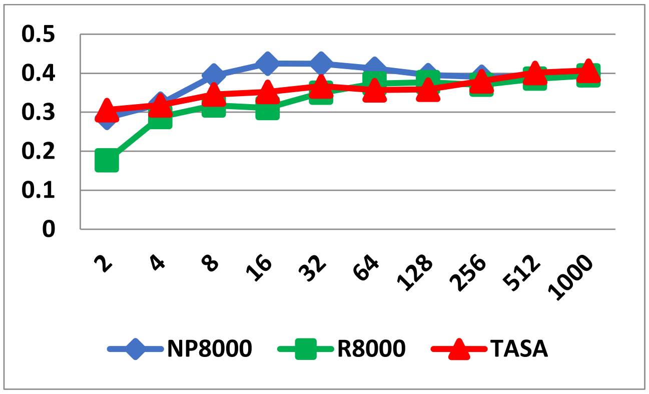


Figure 6. Comparing NP8000, R8000 and TASA on hint responses.

Figure 7 shows the performance of NP8000, R8000 and TASA spaces on prompt questions. Newtonian physics space NP8000 performed best, especially at around *k*=17. For lower dimensionality, TASA space was slightly better than random space. However, after *k*=32, the random space became slightly better than TASA.

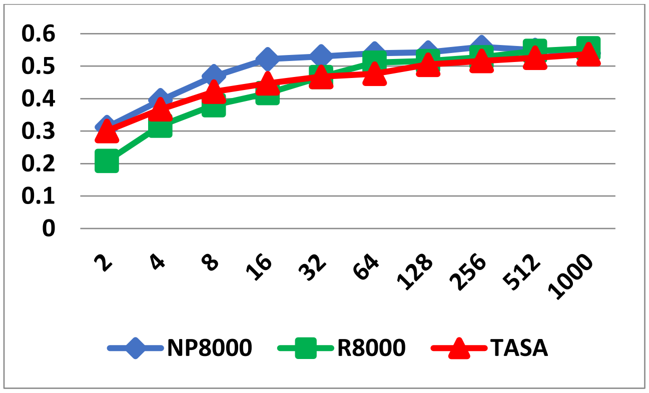


Figure 7. Comparing NP8000, R8000 and TASA space on prompt responses.

4. DISCUSSION AND FUTURE WORK

In AutoTutor applications, LSA similarities has been used as an important feature for building models to evaluate learners' responses. Every time a new application was created, a new "domain specific" spaces was generated. The so called "domain specific" spaces were usually generated from a corpus provided by domain experts. It was often unclear whether or not the documents in the domain corpus were sufficiently representative. That motivated us to explore the impact of document selection and corpus size, taking into account the optimal space dimensionality.

Instead of relying on experts' selections, we used Wikipedia as a universal source to select corpus for any domain. In this study, we used a method called "seeding method" to select Wikipedia articles based on a small seed corpus. Although the seeding method started with an automatic keyness computation and ended with automatic document ranking and selection, the method was not fully automatic, because, in the middle of the process, a manual Wikipedia category selection was involved. Because of this manual selection, the document ranking was constrained by the category selection. That is, the document ranking was computed only over a subset of Wikipedia articles. Although this reduced the searching cost, it is not clear how much better a space could be if the documents were selected from all Wikipedia articles. A fully automatic and inexpensive Wikipedia article selection algorithm apparently is still needed.

The seeding method was not directly evaluated. However, its effect has been shown by the fact that the selected spaces perform significantly better than random spaces. Yet, the seeding method might have room for improvement. Better keyness assignment and document ranking algorithms are possible. For example, the entropy based keyword extraction algorithm provided by Yang et al. [25] is a good candidate for more sophisticated keyness assignment algorithms. Even further, instead of keyness based document ranking, other methods without keyness assignment are possible. For example, a seed LSA space could be generated from the seed corpus. Then a small number of Wikipedia articles could be selected to form a slightly larger corpus. Then a larger LSA space is generated and more Wikipedia articles are added. Such an iterative process could be more expensive but may provide better LSA spaces.

This study revealed several interesting results about the impact of dimensionality. When we examined the performance, we did not expect that *k*=2 could provide a significant correlation. It turned out that 2 dimensional spaces (e.g. NP8000, TASA) could actually perform quite well (around 0.3). This fact is important because two dimensional vectors are easy for visualization. Therefore, if a 2-dimensional space could provide acceptable performance, it may be considered if visualization is a concern.

Another interesting finding is that the optimal *k* could be very small (e.g., 17 for NP8000). A small *k* implies low cost in both storage and computation. However, it may not be possible to identify the optimal *k* without dependent data, such as human ratings. When such data is not available, we certainly want to know what *k* is safe for use. This study showed that *k*=300 is a safe dimensionality for both hint and prompt response evaluation.

It seems obvious that there must be an optimal corpus size, which is not too small and not too large. If a corpus is too small, it may have two problems: 1) it cannot represent the desired domain and 2) it cannot provide enough semantic associations for generating meaningful vectors. If a corpus is too large, it will lose focus. This study shows that NP8000 is better than smaller and larger corpora. The problem, however, is that this optimal size is identified using human rated data. When human ratings are not available, a relatively large corpus would be safer.

TASA space has been widely used in LSA research, as discussed earlier. However, it has been an open question whether a domain specific LSA space would have better performance than a broader TASA space. This study shows that the performance of TASA space on AutoTutor tasks is close to random Wikipedia spaces of large enough corpora and high dimensionality. Even compared with well selected corpora, a TASA space with high dimensionality (e.g., *k*>=300) performs reasonably well on AutoTutor tasks. Therefore, for an application in English language, it should be safe to use TASA space (with *k*=300) when LSA is used for semantic comparison.

However, there are two problems in using TASA. The first problem is that there could be important domain specific terms that are not included in TASA corpus. Another problem is that TASA is an

English corpus. When a space for another language, such as Chinese, French, etc., is needed, there is no simple way to compose a TASA corpus in other languages. Sampling documents from Wikipedia using the seeding method is a good solution to these problems.

There are hundreds of Wikipedias in different languages. Sampling a Wikipedia corpus in any language is easy and free. The seeding method guarantees that the selected articles would include the keywords in the targeted domain. The seeding method also provides significantly better performance with relatively smaller spaces, a smaller vocabulary and a smaller number of dimensions. This means the that the seeding method helps reducing the cost of storage and computing while maintaining performance levels.

In our study, the performance of domain specific spaces could be approximated reached in random Wikipedia spaces or TASA space. The difference is that, domain specific spaces could perform well with very low dimensionality, while non-domain specific spaces need much higher dimensionality to get to the same level of performance. Therefore, the value of using domain specific spaces could be the possible use of low dimensionality. This may have important implications is other applications. For example, in deep learning on natural language processing, reliable low dimensional word embedding will save training cost and make trained models more generalizable.

To conclude, using seeding method and Wikipedia in LSA space generation has the following advantages:

- It guarantees domain keyword inclusion;
- The same method can be applied to all languages;
- It reduces cost of storage and computing; and
- It improves semantic evaluation accuracy.

Once again, LSA similarity is only one of the factors considered in evaluating AutoTutor responses. The correlation values, about 0.43 on hint responses and 0.56 on prompt responses, are still far away from human's agreement (*r*>=0.82). In order to further improve AutoTutor assessment accuracy, other evaluation methods are needed, such as regular expressions. Cai et al. (2016) [4] proposed an alternative way in computing the LSA similarity. Instead of comparing the responses with the author-prepared ideal answer, they compared it with group responses. As we mentioned earlier, combining regular expression with LSA would make a better assessment model. In other words, LSA similarity may be used as a very powerful predictor to build a model to simulate human rating. However, using LSA alone is usually not enough. LSA vectors could also be used as word embedding to train deep learning models [20]. We did not include such algorithms in this paper, because our focus is on the quality of spaces, not the quality of AutoTuotor assessment model.

5. ACKNOWLEDGMENTS

The research on was supported by the National Science Foundation (DRK-12-0918409, DRK-12 1418288), the Institute of Education Sciences (R305C120001), Army Research Lab (W911INF-12-2-0030), and the Office of Naval Research (N00014-12-C-0643; N00014-16-C-3027). 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, IES, or DoD. The Tutoring Research Group (TRG) is an interdisciplinary research team comprised of researchers from psychology, computer science, and other departments at University of Memphis (visit http://www.autotutor.org).

6. REFERENCES

[1] Blei, D.M., Ng, A.Y. and Jordan, M.I. 2015. Latent Dirichlet Allocation. *Journal of Machine Learning Research*. 3, (2015), 993–1022.

[2] Bradford, R.B. 2008. An empirical study of required dimensionality for large-scale latent semantic indexing applications. *Proceeding of the 17th ACM conference on Information and knowledge mining - CIKM '08*. (2008), 153.

[3] Buckley, C. 1985. Implementation of the smart information retrieval system. Ithaca.

[4] Cai, Z., Gong, Y., Qiu, Q., Hu, X. and Graesser, A. 2016. Making autotutor agents smarter: Autotutor answer clustering and iterative script authoring. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 10011 LNAI, (2016), 438–441.

[5] Cai, Z., Graesser, A., Forsyth, C., Burkett, C., Millis, K., Wallace, P., Halpern, D. and Butler, H. 2011. Trialog in {ARIES}: User input assessment in an intelligent tutoring system. *Proceedings of the 3rd IEEE International Conference on Intelligent Computing and Intelligent Systems*. March 2016 (2011), 429–433.

[6] Crossley, S.A., Dascalu, M. and Mcnamara, D.S. 2017. How important is size? An Investigation of Corpus Size and Meaning in both Latent Semantic Analysis and Latent Dirichlet Allocation. *FLAIRS Conference* (2017), 293–296.

[7] Davies, M. 2010. The Corpus of Contemporary American English as the first reliable monitor corpus of English. *Literary and Linguistic Computing*. 25, 4 (2010), 447–464.

[8] Gotoh, Y. and Renals, S. 1997. Document space models using latent semantic analysis. 7, (1997), 6–9.

[9] Graesser, A.C., Feng, S. and Cai, Z. 2017. Two Technologies to Help Adults with Reading Difficulties Improve their Comprehension. *Developmental perspectives in written language and literacy. In honor of Ludo Verhoeven*. E. Segers and P. van den Broek, eds. John Benjamin Publishing Company. 296–313.

[10] Graesser, A.C., Forsyth, C.M. and Foltz, P. 2017. Assessing conversation quality, reasoning, and problem-solving performance with computer agents. *The nature of problem solving: Using research to inspire 21st century learning*. 245–261.

[11] Graesser, A.C., McNamara, D.S., Louwerse, M.M. and Cai, Z. 2004. Coh-Metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, & Computers*. 36, 2 (2004), 193–202.

[12] Hotelling, H. 1953. New Light on the Correlation Coefficient and its Transforms. *Journal of the Royal Statistical Society. Series B (Methodological)*. 15, 2 (1953), 193–232.

[13] Kontostathis, A. 2007. Essential Dimensions of Latent Semantic Indexing (LSI). (2007), 1–8.

[14] Landauer, T.K. and Dumais, S.T. 1997. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *Psychological Review*. 104, 2 (1997), 211–240.

[15] Landauer, T.K., Folt, P.W. and Laham, D. 1998. An introduction to latent semantic analysis. *Discourse processes*. 25, 2 (1998), 259–284.

[16] Martin, D.I., Bear, S. and Consulting, T. 1994. Mathematical Foundations Behind Latent Semantic Analysis. 35–55.

[17] McNamara, D., Cai, Z. and Louwerse, M. 2007. Optimizing LSA Measures of Cohesion. *Handbook of latent semantic analysis*. T.K. Landauer, D.S. McNamara, S. Dennis, and W. Kintsch, eds. Erlbaum. 379–400.

[18] Penumatsa, P., Ventura, M., Graesser, A.C., Louwerse, M., Hu, X., Cai, Z. and Franceschetti, D.R. 2006. The Right Threshold Value: What is the Right Threshold of Cosine Measure When Using Latent Semantic Analysis for Evaluating Student Answers? *International Journal on Artificial Intelligence Tools*. 15, 05 (2006), 767–777.

[19] Peter W. Foltz Lynn A. Streeter, K.E.L.T.K.L. Implementation and Applications of the Intelligent Essay Assessor.

[20] Riordan, B., Horbach, A., Cahill, A., Zesch, T. and Lee, C.M. 2017. Investigating neural architectures for short answer scoring. *$Bea17*. (2017), 159–168.

[21] Slater, S., Joksimović, S., Kovanovic, V., Baker, R.S. and Gasevic, D. 2017. Tools for Educational Data Mining: A Review. *Journal of Educational and Behavioral Statistics*. 42, 1 (2017), 85–106.

[22] Ștefănescu, D., Banjade, R. and Rus, V. 2014. Latent Semantic Analysis Models on Wikipedia and TASA. *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14)*. (2014), 1417–1422.

[23] Steiger, J.H. 1980. Tests for comparing elements of a correlation matrix. *Psychological Bulletin*.

[24] Wiemer-Hastings, P., Graesser, A.C., Harter, D. and Grp, T.R. 1998. The foundations and architecture of autotutor. *Intelligent Tutoring Systems*. 1452, (1998), 334–343.

[25] Yang, Z., Lei, J., Fan, K. and Lai, Y. 2013. Keyword extraction by entropy difference between the intrinsic and extrinsic mode. *Physica A*. 392, 19 (2013), 4523–4531.

Machine Beats Human at Sequencing Visuals for Perceptual-Fluency Practice

Ayon Sen (asen6@wisc.edu)[1], Purav Patel[2], Martina A. Rau (marau@wisc.edu)[2],
Blake Mason[3], Robert Nowak[3], Timothy T. Rogers[4], Xiaojin Zhu[1]
[1] Department of Computer Sciences, [2] Department of Educational Psychology,
[3] Department of Electrical and Computer Engineering, [4] Department of Psychology
University of Wisconsin-Madison

ABSTRACT

In STEM domains, students are expected to acquire domain knowledge from visual representations that they may not yet be able to interpret. Such learning requires perceptual fluency: the ability to intuitively and rapidly see which concepts visuals show and to translate among multiple visuals. Instructional problems that engage students in nonverbal, implicit learning processes enhance perceptual fluency. Such processes are highly influenced by sequence effects. Thus far, we lack a principled approach for identifying a sequence of perceptual-fluency problems that promote robust learning. Here, we describe a novel educational data mining approach that uses machine learning to generate an optimal sequence of visuals for perceptual-fluency problems. In a human experiment, we show that a machine-generated sequence outperforms both a random sequence and a sequence generated by a human domain expert. Interestingly, the machine-generated sequence resulted in significantly lower accuracy during training, but higher posttest accuracy. This suggests that the machine-generated sequence induced desirable difficulties. To our knowledge, our study is the first to show that an educational data mining approach can induce desirable difficulties for perceptual learning.

Keywords

visuals, perceptual fluency, implicit learning, desirable difficulties, machine learning, machine teaching, chemistry, optimal training, sequence effects

1. INTRODUCTION

Visual representations are ubiquitous instructional tools in science, technology, engineering, and math (STEM) domains [2, 23]. For example, chemistry instruction on bonding typically includes the visuals shown in Figure 1. While we typically assume that such visuals help students learn because they make abstract concepts more accessible, they can also impede students' learning if students do not know how the visuals show information [27]. To successfully use visuals to learn new domain knowledge, students need representational competencies: knowledge about how visual representations show information [1]. For example, a chemistry student needs to learn that the dots in the Lewis structure in Figure 1(a) show electrons and that the spheres in the space-filling model in Figure 1(b) show regions where electrons likely reside.

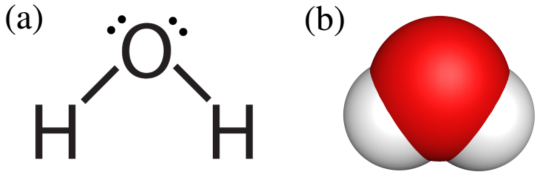


Figure 1: Two commonly used visual representations of water (a: Lewis structure; b: space-filling model).

Most instructional interventions that help students acquire representational competencies focus on *conceptual* representational competencies. These include the ability to map visual features to concepts, support conceptual reasoning with visuals, and choose appropriate visuals to illustrate a given concept [5]. For example, chemists can explain how the number of lines and dots shown in the Lewis structure relate to the colored spheres in the space-filling model by relating these visual features to chemical bonding concepts. Such conceptual representational competencies are acquired via explicit, verbally mediated learning processes that are best supported by prompting students to explain how visuals show concepts [20, 27].

Less research has focused on a second type of representational competency — *perceptual fluency.* It involves the ability to rapidly and effortlessly see meaningful information in visual representations [12, 14]. For example, chemists immediately see that both visuals in Figure 1 show water without having to effortfully think about what the visual shows. They are as fluent at seeing meaning in multiple visuals as bilinguals are fluent in hearing meaning in multiple languages. Perceptual fluency frees up cognitive resources for higher-order complex reasoning, thereby allowing students to use visuals to learn new domain knowledge [16, 27].

Students acquire perceptual fluency via implicit inductive processes [12, 14]. These processes are nonverbal because verbal reasoning is not necessary [19] and may even interfere with the acquisition of perceptual fluency [20]. Consequently, instructional problems that enhance perceptual fluency engage students in simple problems to quickly judge what a visual shows [19]. For example, one type of perceptual-fluency problem may ask students to quickly and intuitively judge whether two visuals like the ones in Figure 1 show the same molecule. They ask students to rely on implicit intuitions when responding to a series of perceptual-fluency problems. Students typically receive numerous perceptual-fluency problems in a row. The problem sequence is typically chosen so that (1) students are exposed to a variety of visuals and (2) consecutive visuals vary incidental features while drawing students' attention to conceptually relevant features [19, 27].

However, these general principles are underspecified in the sense that they leave room for many possible problem sequences. To date, we lack a principled approach capable of identifying sequences of visual representations that yield optimal learning outcomes for perceptual-fluency problems. To address this issue, we developed a novel educational data mining approach. Using data from human students who learned with perceptual-fluency problems, we trained a machine learning algorithm to mimic human perceptual learning. Then, we used an algorithm to search over possible sequences of visual representations to identify the sequence that was most effective for a machine learning algorithm. In a human experiment, we then tested whether (1) the machine-selected sequence of visual representations yielded higher learning outcomes compared to (2) a random sequence and (3) a sequence generated by a human expert based on perceptual learning principles.

In the following, we first review relevant literature on learning with visual representations, perceptual fluency, and our machine learning paradigm. Then, we describe the methods we used to identify the machine-selected sequence and the methods for the human experiment. We also discuss how our results may guide educational interventions for representational competencies and educational data mining more broadly.

2. PRIOR RESEARCH

2.1 Learning with Visual Representations

Theories of learning with visual representations define visual representations as a specific type of external representation. External representations are objects that stand for something other than themselves — a referent [25]. When we see an image of a pizza, for example, the referent could be a slice of pizza (a concrete object). Alternatively, when used in the context of math instruction, the referent could be a fraction of a whole pizza (an abstract concept). Representations used in instructional materials are defined as external representations because they are external to the viewer. By contrast, internal representations are mental objects that students can imagine and mentally manipulate. Internal representations are the building blocks of mental models; these models constitute students' content knowledge of a particular topic or domain. External representations can be symbolic or visual. For instance, text or equations are symbolic external representations that consist of symbols that have arbitrary (or convention-based) mappings to the referent [32]. By contrast, *visual representations* have similarity-based mappings to the referent [32].

Several theories describe how students learn from visual representations. Mayer's [22] Cognitive Theory of Multimedia Learning (CTML) and Schnotz's [32] Integrated Model of Text and Picture Comprehension (ITPC) draw on information processing theory [4] to describe learning from external representations as the integration of new information into a mental model of the domain knowledge. Here, we focus on learning processes relevant to visual representations.

First, students select relevant *sensory information* from the visual representations for further processing in working memory. To this end, students use perceptual processes that capture visuo-spatial patterns of the representation in working memory [32]. To willfully direct their attention to relevant visual features, students draw on conceptual competencies that enable top-down thematic selection of visual features [15, 17].

Second, students *organize* this information into an internal representation that describes or depicts the information presented in the external representation. Because visual representations have similarity-based analog mappings to referents, their structure can be directly mapped to the analog internal representations [10, 32]. In forming the internal representation, students engage perceptual processes that draw on pattern recognition of objects based on visual cues. They engage conceptual processes to map the visual cues to conceptual representational competencies that allow the retrieval of concepts associated with these objects. The resulting internal representation is a perceptual analog of the visual representation. It is depictive in that its organization directly corresponds to the visuo-spatial organization of the external visual representation [32].

Third, students integrate the information contained in the internal representations into a *mental model* of the domain knowledge (e.g., schemas, category knowledge). To this end, students integrate the analog internal representation into a mental model by mapping the analog features to information in long-term memory. This third step is what constitutes learning: students learn by integrating internal representations into a coherent mental model of the domain knowledge [22, 32, 37].

In sum, students' learning from visual representations hinges on their ability to form accurate internal representations of the representations' referents and on their ability to integrate internal representations into a coherent mental model of the domain knowledge. This process involves both conceptual and perceptual competencies [27]. Although it is well established that conceptual and perceptual competencies are interrelated [16, 17], it makes sense to distinguish them because they are acquired via qualitatively different learning processes [16, 19, 20]. As mentioned earlier, conceptual representational competencies are acquired via verbally mediated, explicit processes [20, 27]. By contrast, perceptual fluency is acquired via implicit, mostly nonverbal processes. Whereas most prior research on instructional interventions for representational competencies has focused on conceptual processes, we focus on perceptual processes.

2.2 Perceptual Fluency

Research on perceptual fluency is based on findings that experts can automatically see meaningful connections among representations, that it takes them little cognitive effort to translate among representations, and that they can quickly and effortlessly integrate information distributed across representations [12]. For example, experts can see “at a glance” that the Lewis structure in Figure 1(a) shows the same molecule as the space-filling model in Figure 1(b). Such perceptual fluency frees cognitive resources for explanation-based reasoning [14,31] and is considered an important goal in STEM education.

According to the CTML and the ITCP, perceptual fluency involves efficient formation of accurate internal representations of visual representations [22,32]. Perceptual fluency also involves the ability to combine information from different visual representations without any perceived mental effort and to quickly translate among them [7] [19]. According to the CTML and ITCP, this allows students to map analog internal representations of multiple visual representations to one another [22,32].

Cognitive science literature [12, 15, 20] suggests that students acquire perceptual fluency via perceptual-induction processes. These processes are inductive because students can infer how visual features map to concepts through experience with many examples [12,15,19]. Students gain *efficiency* in seeing meaning in visuals via perceptual chunking. Rather than mapping specific analog features to concepts, students learn to treat each analog visual as one perceptual chunk that relates to multiple concepts. Perceptual-induction processes are thought to be nonverbal because they do not require explicit reasoning [20]. They are implicit because they occur unintentionally and sometimes unconsciously [33].

Interventions that target perceptual fluency are relatively novel. Kellman and colleagues [19] developed interventions that engage students in perceptual-induction processes by exposing them to many short problems where they have to rapidly translate between representations. For example, students might receive numerous problems that ask them to judge whether two visuals like the ones shown in Figure 1 show the same molecule. These interventions have enhanced students’ learning in domains like chemistry [30,36].

Perceptual learning is strongly affected by problem sequences [27]. To design appropriate problem sequences, consecutive problems expose students to systematic variation (often in the form of contrasting cases) so that irrelevant features vary but relevant features appear across several problems [19]. However, a vital issue remains when designing problem sequences for perceptual-fluency problems: Visual representations differ on a large number of visual features. Consequently, countless potential problem sequences exist that systematically vary these visual features. How do we know which sequence is most effective? To address this issue, we propose a new educational data mining approach that draws on Zhu’s machine-teaching paradigm [38,39]

2.3 Machine Teaching Paradigm

Simply put, machine teaching is the inverse problem of machine learning. Machine learning refers to computer algorithms that select an optimal model for a given set of data. In other words, it determines which model fits the data best. Machine teaching, on the other hand, finds the optimal (smallest) set of data for training such that a given algorithm selects a target model. Although the machine teaching paradigm has been applied to cognitive psychology and education [24], it has not yet been used in educational data mining research.

Machine teaching requires a cognitive model i.e.,a learning algorithm that mimics how human students learn a mapping between visual representations like the ones shown in Figure 1). Given the cognitive model, machine teaching seeks a sequence of learning problems (optimal training sequence $\mathcal{O}$) such that when given $\mathcal{O}$, the learning algorithm learns the mapping. Here, $\mathcal{O}$ need not be independent and identically distributed (i.i.d.). Machine teaching can be viewed as a communication problem between a teacher and a student: The goal is to communicate the mapping using the shortest message. The channel only allows messages in the form of a training sequence and the student decodes the message with the learning algorithm. In perceptual learning, students learn a mapping between visual features of two types of visual representations, allowing them to fluently translate among the visual representations.

To evaluate whether a training sequence is effective, we test the cognitive model’s performance at mapping visual representations using a different set of perceptual-fluency problems than used during training. Typically, a sequence of training problems (aka training instances in machine learning) is drawn from a distribution of perceptual-fluency problems used for training (P_t). The set of test problems comes from a separate distribution of perceptual-fluency problems (P_e). The goal is to minimize the test error rate on P_e. The goal of machine teaching then becomes:

$$\mathcal{O} = \operatorname*{argmin}_{S \in \mathcal{C}_t} P_{(x,y)\sim P_e}\left(\mathcal{A}(S)(x) \neq y\right) \tag{1}$$

Here, $\mathcal{C}_t$ is the set of all possible training sequences and $\mathcal{A}(S)$ is the learned hypothesis after training on the sequence S. Note that, $\mathcal{O}$ is not necessarily an i.i.d. sequence drawn from P_t. One practical approach to approximately solve the optimization problem is shown in Algorithm 1. To properly construct the optimal training sequence in this given setting, we must understand:

1. the nature of the to-be-learned domain knowledge
2. the learning algorithm the cognitive model is using

In this paper, the to-be-learned domain knowledge is well-known. It is the mappings between visual representations that students have to learn. Further, we used data from human students learning from perceptual-fluency problems to generate a cognitive model that mimics how humans learn mappings between visual representations. Our goal is to investigate whether, when the mappings and the cognitive model are well understood, machine teaching can identify a training set that is more effective than (a) a problem sequence based on perceptual learning principles and (b) a random sequence.

Algorithm 1 Machine Teaching

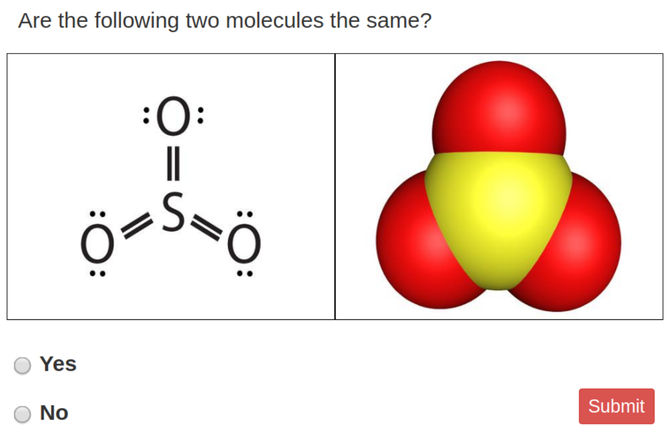
```
Input: Learner $\mathcal{A}$, Test Distribution $P_e$
$\mathcal{O} \leftarrow$ Starting sequence
$\epsilon_{\text{best}} \leftarrow$error(train($\mathcal{A}, \mathcal{O}$), $P_e$)
while TRUE do
    $\mathcal{N} \leftarrow$neighbors($\mathcal{O}$), $\epsilon_{\text{old}} \leftarrow \epsilon_{\text{best}}$
    for $S \in \mathcal{N}$ do
        $\epsilon \leftarrow$error(train($\mathcal{A}, S$), $P_e$)
        if $\epsilon < \epsilon_{\text{best}}$ then
            $\epsilon_{\text{best}} \leftarrow \epsilon, \mathcal{O} \leftarrow S$
        end if
    end for
    if $\epsilon_{\text{best}} = \epsilon_{\text{old}}$ then
        return $\mathcal{O}$
    end if
end while
```

3. COGNITIVE MODEL

We now describe how we constructed the cognitive model that was used to construct the training sequence. To this end, we first describe the perceptual-fluency problems, then describe how we formally represented these problems, which learning algorithm the cognitive model used, and finally how we used the cognitive model to identify the optimal training sequence.

3.1 Perceptual-Fluency Problems

Perceptual-fluency problems are single-step problems that ask students to make simple perceptual judgments. In our case, students were asked to judge whether two visual representations showed the same molecule, as shown in Figure 2. Students were given two images. One image was of a molecule represented by a Lewis structure and the other image was a molecule represented by a space-filling model. They were asked to judge whether those two images show the same molecule or not.



Are the following two molecules the same?

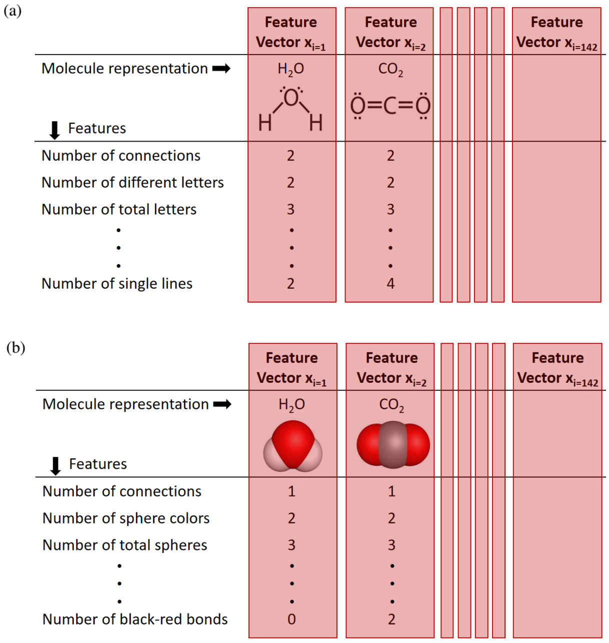
○ Yes

○ No

Figure 2: In this sample perceptual-fluency problem, students judged whether or not the Lewis structure and the space-filling model showed the same molecule. The answer is yes.

3.2 Visual Representation of Molecules

In our experiment, we used visual representations of chemical molecules common in undergraduate instruction. To identify these molecules, we reviewed textbooks and web-based instructional materials. We counted the frequency of different molecules using their chemical names (e.g., H_2O) and common names (e.g., water), and chose the 142 most common molecules. In order to formally describe the visual representations, we quantified visual features for each of the molecules. To this end, we first hand-coded the visual features that were present in the visual representations. For Lewis structures, these hand-coded features included counts of individual letters as well as information about different bonds present in each molecule, among others. For space-filling models, hand-coded features included counts of colored spheres, bonds, and other features. Further, we included several surface features that we expect human students attend to based on findings that humans tend to focus on broader surface features that are easily perceivable. Then we used the method found in [29] to determine which subset of features (each for Lewis structure and space-filling model) humans attend to most. Building on these results, we created feature vectors for each of the molecules (Figure 3). These feature vectors of Lewis structures and space-filling models contained 27 and 24 features, respectively. These feature vectors were then used to train and test the learning algorithm.



(a)

	Feature Vector $x_{i=1}$	Feature Vector $x_{i=2}$	Feature Vector $x_{i=142}$
Molecule representation ➡	H_2O	CO_2	
⬇ Features			
Number of connections	2	2	
Number of different letters	2	2	
Number of total letters	3	3	
⋮	⋮	⋮	
Number of single lines	2	4	

(b)

	Feature Vector $x_{i=1}$	Feature Vector $x_{i=2}$	Feature Vector $x_{i=142}$
Molecule representation ➡	H_2O	CO_2	
⬇ Features			
Number of connections	1	1	
Number of sphere colors	2	2	
Number of total spheres	3	3	
⋮	⋮	⋮	
Number of black-red bonds	0	2	

Figure 3: Example features for H_2O and CO_2 molecule representations with feature vectors in red (a: Lewis structure; b: space-filling model).

3.3 Learning Algorithm

We used a feed-forward artificial neural network (ANN) [8] as our learning algorithm. ANN is inspired by the biological neural network. A biological neuron produces an output when collective effect of its inputs reaches a certain threshold. It is still not clear exactly how the human brain learns but one assumption is that it is associated with the interconnection between the neurons. ANNs try to model this low level functionality of the brain. We chose ANN to be our learning algorithm due to this similarity. Our ANN took two feature vectors (x_1 and x_2) as input. Each feature vector corresponded to one of the two molecules shown. Given this input, the ANN produced a probability that the two molecules were the same. Then, given the correct answer $y \in \{0, 1\}$ (here 1 means the two molecules are the same), the ANN updated its weights using the backpropagation algorithm. The backpropagation algorithm uses gradients to converge to an optima. Algorithm 2 shows the training procedure of the neural network. It shows that the update procedure also used a history window and multiple backpropagation passes, an atypical approach for an ANN. We took two measures to address the issue that regular ANN algorithms do not learn from memory like humans do. First, we assumed that humans remember a fixed number of past consecutive problems. Second, we assumed that after receiving feedback on the latest problem, humans update their internal model by reviewing memorized problems (along with the latest problem) several times. To emulate this behavior, we introduced the history window and multiple backpropagation passes. This procedure was followed for all problems in a given training sequence.

Algorithm 2 train: training method for the NN learner

```
Input: Training sequence $S$ , Learning rate $\eta$, History window size $w$, Number of backpropagations $b$
$H \leftarrow []$ //initialize an empty history window
for $i = 1 \to |S|$ do
    append($H, S[i]$) //update history window
    // train on the history window
    $w' \leftarrow |H|$
    for $k = 1 \to b$ do
        for $j = 1 \to w'$ do
            $(\mathbf{x}, y) \leftarrow H[j]$
            backprop($\mathbf{x}, y, \eta$)
        end for
    end for
    //check history window size
    if $w' > w$ then
        $H$.remove(0) //remove the oldest instance in history
    end if
end for
```

A further, structural difference between our learning algorithm from a general artificial neural network is that our learning algorithm had two separate weight columns (one for each representation of the input molecules). The model architecture of the ANN is shown in Figure 4. Here, the weights and outputs from one of the columns did not interact with those of the other column until the output layer. The network mapped the two inputs (feature vectors x_1 and x_2) to a space wherein the same molecule shown by different representations are close to each other while different molecules are distant. These mapping functions are called embedding functions (one for each representation) and the space is called a common embedding space. Once the mapping was complete, a judgment was possible regarding the similarity of the input molecules. This judgment was based on the distance in the common embedding space and made in the output layer of the ANN. Embeddings were generated in the layer before the output layer—the embedding layer.

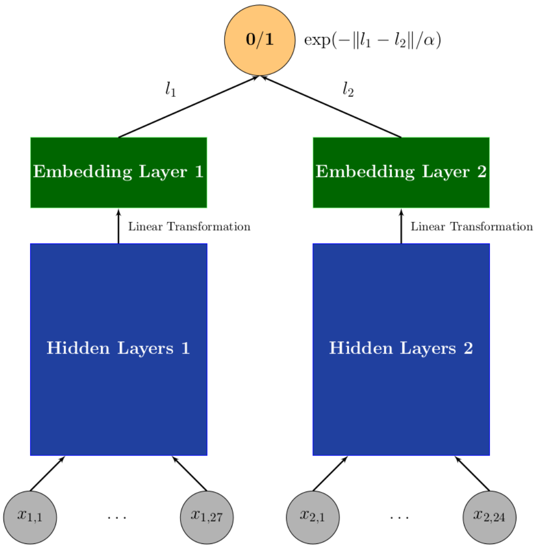


Figure 4: Structure of the Artificial Neural Network learning algorithm

Neurons in an ANN use a non-linear function called activation function to introduce non-linearity. For all hidden layers before the embedding layer, we used the leaky rectifier [21] activation function (the neuron employing leaky rectifier is called a leaky rectified linear unit or leaky ReLU). A standard rectified linear unit (ReLU) allows only positive inputs to move onwards (outputs 0 otherwise). A leaky ReLU, on the other hand, outputs a small scaled input when the input is negative. Both ReLU and leaky ReLU have strong biological motivations. According to cognitive neuroscience studies of human brains, neurons encode information in a sparse and distributed fashion [3]. Using ReLU, ANNs can also encode information sparsely. Besides this biological plausibility, sparsity also confers mathematical benefits like information disentangling and linear separability. Rectified linear units also enable better training of ANNs [13]. The embedding layers, by contrast, do not use activation functions. Hence, the output of embedding layers are a linear transformation of its inputs. Given the inputs (x_1, x_2), let the ANN-generated embeddings be l_1 and l_2, respectively. Then, we computed the probability of the two representations showing the same molecule in the output layer using the following equation:

$$\exp\left(-\frac{\|l_1 - l_2\|}{\alpha}\right) \tag{2}$$

Here, α is a trainable parameter that the ANN learns along with the weights. We thresholded this value at 0.5 to generate the ANN prediction $\hat{y} \in \{0, 1\}$.

3.4 Pilot Study - Train the Learning Algorithm

Our first step was to train the learning algorithm to mimic human perceptual learning. To this end, we conducted a pilot experiment to find a good set of hyperparameters for the ANN learning algorithm. Hyperparameters of an ANN are variables that are set before optimizing the weights (e.g., number of hidden layers, number of neurons in each layer, learning rate etc.). Our goal was to identify hyperparameters that make predictions matching human behavior on the posttest. Hence, we matched the algorithm's predictions to summary statistics of human performance on the posttest.

Our pilot experiment included 47 undergraduate chemistry students. They were randomly assigned to one of two conditions that used a random training sequence: supervised training ($n = 35$) or unsupervised training ($n = 12$). Participants in the supervised training condition received feedback after each training problem, whereas participants in the unsupervised condition did not receive feedback. We included the unsupervised training condition to generate an evaluation set (used to determine the success of pretraining). This evaluation set was used to pretrain the ANN learning algorithm.

Let there be n supervised human participants. Each participant received a random pretest set, a random training sequence, and a random posttest set. We trained the ANN learning algorithm n times independently (once for each participant). While training for the i-th time we used the training sequence viewed by the i-th supervised human participant. The same posttest set viewed by this participant was also used to evaluate the performance of the ANN learning algorithm after training. Let the error on this posttest set for the i-th human participant and trained ANN learning algorithm be pp_i and pn_i respectively. Then, Equation 3 is a measure used to determine whether or not an ANN learning algorithm's performance is comparable to the average human. Note that lower *error rates* are desirable.

$$\textit{error rates} = \left| \frac{1}{n} \left(\sum_{i=1}^{n} pp_i - \sum_{i=1}^{n} pn_i \right) \right| \tag{3}$$

Table 1 reports the accuracies of participants in the pilot experiment.

Table 1: Accuracy in Pilot Experiment by Training Condition. Average pretest, training and posttest accuracy with SEM in parentheses.

Condition	Pretest	Training	Posttest
Supervised	79.9 (1.8)	75.7 (1.2)	89.4 (1.4)
Unsupervised	77.9 (3.4)	78.5 (2.8)	77.1 (3.3)

We note that humans usually have some degree of prior knowledge about chemistry. By contrast, the weights of an ANN are generally initialized at random. We address this issue by modeling the effect of prior knowledge, specifically we introduced a pretraining phase for the ANN learning algorithm. To this end, we drew a large sample of instances (10000) from the combined test and training distribution ($\frac{1}{2}P_e + \frac{1}{2}P_t$) to form a pretraining set. Further, we combined the pretest problem across both the supervised and unsupervised conditions, along with the training problems in the unsupervised condition to form the pretraining evaluation set. Because we did not provide feedback for these problems, we assumed that the participants did not learn anything new while going through them. Formally, let participants' error on the pretraining evaluation set be called human pretraining error. We then trained the ANN learning algorithm on the pretraining set. Note that an ANN can train over the over the same set over multiple iterations (formally known as epochs). We trained the ANN learning algorithm until its error on the pretraining evaluation set was smaller than human pretraining error. This concluded the pretraining phase.

We used standard coordinate descent with random restart to find a good hyperparameter set. Coordinate descent successively minimizes the *error rates* along the coordinate directions (e.g., embedding size, learning rate). At each iteration, the algorithm chooses one particular coordinate direction while fixing the other values. Then, it minimizes in the chosen coordinate direction. Table 2 shows the values of the hyperparameters over which we decided to explore along with the best value found. These hyperparameters were used to identify the optimal training sequence.

3.5 Finding an Optimal Training Sequence

We used the ANN learning algorithm to generate an optimal training sequence for the perceptual-fluency problems. In Equation 1, we defined the optimization problem to solve. We solved this problem by searching over the space of all possible training sequences. Without limiting the size of the training sequence, the search space becomes infinite and infeasible. To mitigate this issue, we set the size of the candidate training sequences to 60. This aligns with prior research on perceptual learning [28]:

$$\mathcal{O} = \underset{S \in \mathcal{C}_t, |S|=60}{\operatorname{argmin}} \; P_{(x,y)\sim P_e} \left(\mathcal{A}(S)(x) \neq y \right) \tag{4}$$

We used a modified hill climbing algorithm to find such an optimal training sequence. Hill climb search takes a greedy approach. Procedurally, we started with one particular training sequence. Then, we evaluated neighbors of that particular training sequence to determine whether a better one existed. If so, we moved to that one. This process stopped when no such neighbors were found. This search algorithm is defined with its states and neighborhood definition:

- **States:** Any training sequence $S \in \mathcal{C}_t$ of size 60
- **Initial State:** A training sequence selected by a domain expert.
- **Neighborhood of S:** Any training sequence that differs with S by one problem is a neighbor. For computational efficiency, we restricted ourselves to only inspecting 500 neighbors for a given training sequence. We do so by first selecting a problem S uniformly at random. Then we replace the selected problem with 500 randomly selected problems with the same answer (i.e., same y value). This made our search algorithm stochastic.

4. HUMAN EXPERIMENT

Our main goal was to evaluate whether the optimal training sequence yields higher learning outcomes. To this end, we conducted a randomized, controlled experiment with humans. Here, we discuss our experimental setup and associated results.

Table 2: Hyper-parameters for the ANN learning algorithm

Parameter name	**Values explored**	**Best value**
Embedding size	$1, 2, 4, 8, 16$	2
Learning rate	$0.00001, 0.00005, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1$	0.0001
History window size	$0, 1, 2, 4, 8, 16, 32, 60$	2
Backprop count	$1, 2, 4, 8, 16$	2
Number of hidden layers before embedding layer	$0, 1, 2, 3, 4$	0
Number of hidden units in each column	$10, 20, 40, 80, 160$	N/A

4.1 Participants

We recruited 368 participants using Amazon's Mechanical Turk (MTurk) [6]. Among them, 216 were male and 131 were female. The rest did not disclose their gender. Most of the participants were below the age of 45 (86%) and the greatest number (192) fell in the age group $24 - 35$. Among the 95.4% who disclosed their knowledge about chemistry, 45.7% had taken an undergraduate-level chemistry class.

4.2 Test Set

Because our goal was to assess transfer of learning from the training sequence to a novel test set, we chose training and test problems from separate distributions. Hence, we randomly divided the 142 molecules that we selected for this experiment into two sets of 71 (training molecules, $\mathcal{X}_t$ and test molecules $\mathcal{X}_e$). One of the sets was used to create the test distribution, whereas the other one was used to create the training distribution. We now describe in more detail how we created the test distribution P_e because our goal was to reduce humans' error rates on the test set. We used the following procedure.

- $x_1 \sim p_1$, where p_1 is a marginal distribution on $\mathcal{X}_e$. p_1 is "importance of molecule x_1 to chemistry education" and was constructed by manually searching a corpus of chemistry education articles for molecule text frequency.
- With probability 1/2, set $x_2 = x_1$ so that the true answer $y = 1$.
- Otherwise, draw $x_2 \sim p_2(\cdot \mid x_1)$. The conditional distribution p_2 is based on domain experts' opinion that favors confusable x_1, x_2 pairs in an education setting. Also note that, $p_2(x_1|x_1) = 0, \forall x_1$. Taken together,

$$P_e(x_1, x_2) = \frac{1}{2} p_1(x_1) \mathbb{I}_{\{x_1 = x_2\}} + \frac{1}{2} p_1(x_1) p_2(x_2 \mid x_1).$$

Both the pretest and posttest judgment problems were sampled from this distribution across all conditions.

4.3 Experimental Design

We compared three training conditions:

1. In the *machine training sequence* condition, we used the optimal training sequence $\mathcal{O}$ found by the modified hill climb search algorithm. For all $(x_1, x_2) \in \mathcal{O}$ (here $x_1 \in \mathcal{X}_t, x_2 \in \mathcal{X}_t$), the corresponding true answer y was the indicator variable on whether x_1 and x_2 were the same molecule: $y = \mathbb{I}_{\{x_1 = x_2\}}$. We presented x_1 and x_2 in Lewis and space-filling representations to the human participants, respectively. Participants gave their binary judgment $\hat{y} \in \{0, 1\}$. We then provided the true answer y as feedback to the participant.
2. In the *human training sequence* condition, the training sequence was constructed by a domain expert using perceptual learning principles (using molecules only from $\mathcal{X}_t$). Specifically, an expert on perceptual learning constructed the sequence based on the contrasting cases principle [19, 30], so that consecutive examples emphasized conceptually meaningful visual features, such as the color of spheres that show atom identity or the number of dots that show electrons. The rest of this condition was the same as the machine training sequence condition. This training sequence is identical to the initial state of the modified hill climb search algorithm that we used to generate the machine training sequence.
3. In the *random training sequence* condition, each training problem (x_1, x_2) was selected from the training distribution P_t with $y = \mathbb{I}_{\{x_1 = x_2\}}$. The training distribution P_t for this condition was induced in the same manner as the test distribution P_e but on the set of training molecules $\mathcal{X}_t$. The rest of the condition was the same as the previous ones.

4.4 Procedure

We hosted the experiment on the Qualtrics survey platform [26] using NEXT [18]. Participants first received a brief description of the study and then completed a sequence of 126 judgment problems (yes or no). The problems were divided into three phases as follows. First, participants received a pretest that included 20 test problems without feedback. Second, participant received the training, which included 60 training problems sequenced in correspondence to their experimental condition. During this phase, correctness feedback was provided for submitted answers. We assumed that participants learned during this phase because they received feedback. Third, participants received a posttest that included 40 test problems without feedback. In addition, one *guard problem* was inserted after every 19 problems throughout all three phases. A guard question either showed two identical molecules depicted by the same representation or two highly dissimilar molecules depicted by Lewis structures. We used these guard questions to filter out participates who clicked through the problems haphazardly. In our main analyses, we disregarded the guard problems. So that no visual representation was privileged, we randomized their positions (left vs. right).

4.5 Results

Of the 368 participants, we excluded 43 participants who failed any of the guard questions. The final sample size was $N = 325$. The final number of participants in the conditions random, human, and machine training sequence were 108, 117 and 100 respectively. Table 3 reports accuracy on the pretest, training set, and posttest. See Figure 5 for a graphical depiction of the same data.

Table 3: Accuracy by Training Condition. Average pretest, training and posttest accuracy with SEM in parentheses.

Condition	Pretest	Training	Posttest
Machine	69.5 (1.1)	63.9 (1.1)	74.7 (1.1)
Human	71.3 (1.3)	72.4 (1.0)	71.7 (1.0)
Random	69.4 (1.1)	70.3 (1.1)	71.1 (1.1)

4.5.1 *Effects of condition on training accuracy*

First, we tested whether training condition affected participants' accuracy during training. To this end, we used an ANCOVA (Analysis of COVAriance) with condition as the independent factor and training accuracy as the dependent variable. Because pretest accuracy was a significant predictor of training accuracy, we included pretest accuracy as the covariate. Results showed a significant main effect of condition on training accuracy, $F(2, 321) = 18.8, p < .001, \eta^2 = .082$. Tukey post-hoc comparisons revealed that (a) the machine training sequence condition had significantly lower training accuracy than the human training sequence condition ($p < .001, d = -0.32$), (b) the machine training sequence condition had significantly lower training accuracy than the random training sequence condition ($p < .001, d = -0.26$), and (c) no significant differences existed between the human and random training sequence conditions ($p = .592, d = 0.05$). In other words, during the training phase, the human and random training sequences were equally effective in terms of accuracy, but the machine training sequence was less effective.

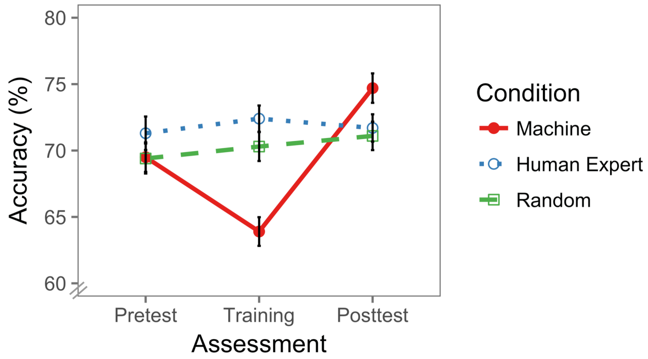


Figure 5: Learning progress between conditions revealed an initial disadvantage, but ultimate advantage for the machine-generated sequence.

4.5.2 *Effects of condition on posttest accuracy*

Next, we tested whether training condition affected participants' posttest accuracy. To this end, we conducted an ANCOVA with condition as the independent factor and posttest accuracy as the dependent variable. Because pretest accuracy was a significant predictor of posttest accuracy, we included pretest accuracy as a covariate. Results showed a significant main effect of condition on posttest accuracy, $F(2, 321) = 5.02, p < .01, \eta^2 = .023$. Tukey post-hoc comparisons revealed that (a) the machine training sequence condition had significantly higher posttest accuracy than the human training sequence condition ($p < .05, d = 0.16$), (b) the machine training sequence condition had significantly higher posttest accuracy than the random sequence condition ($p < .05, d = 0.14$), and (c) no significant differences existed between the human and random training sequence conditions ($p = .960, d = -0.02$). In other words, the human and random training sequences were equally effective and the machine training sequence was most effective.

5. DISCUSSION

Our goal was to investigate whether a novel educational data mining approach can help identify a training sequence of visual representations that enhances students' learning from perceptual-fluency problems. To this end, we applied the machine teaching paradigm. It involved gathering data from human students learning from perceptual-fluency problems. Next, we generated a cognitive model that mimics human perceptual learning. We then used the cognitive model to reverse-engineer an optimal training sequence for a machine-learning algorithm. Finally, we conducted an experiment that compared the machine training sequence to a random sequence and to a principled sequence generated by a human expert on perceptual learning. Results showed that the machine training sequence resulted in lower performance during training, but greater performance on a posttest.

These findings make several important contributions to the perceptual learning literature. First, our results can inform the instructional design of perceptual-learning problems. Even though prior research yields principles for effective sequences of visual representations, numerous potential sequences can satisfy these principles. Our results show that this new educational data mining approach can help address this problem. Given a learning algorithm that constitutes a cognitive model of students learning a task, instructors can identify a sequence of problems that likely yields higher learning outcomes.

Second, our results expand theory on perceptual learning. The fact that the machine learning sequence yielded lower performance during training but greater posttest scores suggests that this sequence induced desirable difficulties during learning [19, 34, 40]. The concept of desirable difficulties describes the common finding that instructional techniques yield lower performance during training, but higher long-term learning outcomes. To explain this phenomenon, Soderstrom and Bjork [34] proposed that more difficult learning interventions induce more active processing during training. This lowers immediate performance due to the increased difficulty, but results in more durable memories and greater long-term learning. Our findings suggest that the machine teaching approach was successful because it identified a training sequence that induced desirable difficulties. To the best of our knowledge, our study is the first to show that an educational data mining approach can be used to induce desirable difficulties for perceptual learning.

Our findings also contribute to the educational data mining literature. We provide the first empirical evidence that an ANN learning algorithm constitutes an adequate cognitive model of learning with visual representations. As far as we know, the machine teaching paradigm has thus far only been applied to learning with artificial visual stimuli that vary on only one or two dimensions (e.g. Gabor patches [11]). Thus, our study provides the first demonstration that machine learning along with machine teaching is a viable approach to modeling and improving learning with realistic, high-dimensional visual representations like Lewis structures and space-filling models of chemical molecules. Many other domains like biology, engineering, math also use high-dimensional visual representations. Therefore, we believe this approach is valuable for educational data mining research.

6. LIMITATIONS AND FUTURE DIRECTIONS

Our findings should be interpreted against the background of the following limitations. First, the population of MTurk workers may limit generalization to the target population of undergraduate chemistry students. MTurk workers have highly variable prior knowledge about chemistry. As mentioned previously, around 45.7% of the participants had taken an undergraduate level chemistry class. This suggests that their prior knowledge may have been lower and more diverse than that of a typical undergraduate chemistry student. Hence, we plan to test whether the machine training sequence leads to better learning for undergraduate chemistry students.

Second, the search algorithm we used to find the machine training sequence did not test all possible training sequences of size 60. As mentioned previously, we only inspected 500 neighbors (out of a potential $5040 = 71 \times 71 - 1$) for any given training sequence. Moreover, we stopped the search algorithm after a predetermined amount of time. We chose this inexhaustive approach because exhaustively finding a solution is not computationally feasible. Thus, we settled for a suboptimal training sequence that still yielded a small risk on the test distribution. Consequently, it is possible to find a better training sequence than the one we used in our experiments.

Third, while determining the hyperparameters of the ANN learning algorithm such that it mimics human perceptual learning, we only searched over a subset of all possible hyperparameters. As a result, it is possible that a better set of hyperparameters exists. Our study was also limited in that we did not account for individual prior knowledge. Hence, future research needs to investigate how to expand the approach presented in this paper to modeling individual prior knowledge (e.g., for adaptive teaching or personal training).

A fourth limitation of the present experiments is that our study was constrained in the use of chemistry representations as stimuli. While we used realistic representations that are more high-dimensional than prior perceptual learning studies [9, 11, 35] and that are more representative of commonly used visual representations in a variety of STEM domains, the complexity of the representations we considered does not reflect all realistic stimuli. Still we see no reason why this approach could not be applied to other representations in other domains. Sparser and richer visuals exist and it is possible that machine teaching may yield greater benefits for sparser visuals. We will investigate this hypothesis in future studies.

7. CONCLUSION

This paper advanced a novel educational data mining approach to identify optimal sequences of visual representations for perceptual-fluency problems. Students' difficulties in learning with visual representations is partly due to a lack of perceptual fluency. This increases the cognitive demands of learning with visuals. Perceptual-fluency problems are a relatively novel type of instructional intervention that can aid learning from visuals by freeing up cognitive resources for higher-order complex reasoning. Thus far, we have lacked a principled approach capable of identifying effective sequences of visual representations. Our educational data mining approach relied solely on students' responses to perceptual-fluency problems to select a sequence of visuals that is effective for a machine learning algorithm mimicking human perceptual learning. Our results showed that this approach is more effective than conventional perceptual fluency instruction. Further, the effectiveness of our approach lies in its ability to induce desirable difficulties. Given the pervasiveness of visual representations in STEM domains, we anticipate that our findings will be broadly useful for students' learning with visual representations. We also plan to investigate how the machine generated sequence induced desirable difficulties in the humans.

Acknowledgement

This is supported in part by NSF grant IIS 1623605. 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 NSF. We also thank Michael Mozer and Brett Roads (University of Colorado, Boulder) for their support and comments regarding the cognitive model.

8. REFERENCES

[1] Ainsworth, S. DeFT: A conceptual framework for considering learning with multiple representations. *Learning and instruction 16*, 3 (2006), 183–198.

[2] Ainsworth, S. The educational value of multiple-representations when learning complex scientific concepts. *Visualization: Theory and practice in science education* (2008), 191–208.

[3] Attwell, D., and Laughlin, S. B. An energy budget for signaling in the grey matter of the brain. *Journal of Cerebral Blood Flow & Metabolism 21*, 10 (2001), 1133–1145.

[4] Baddeley, A. Working memory. *Science 255*, 5044 (1992), 556–559.

[5] Bodemer, D., Ploetzner, R., Feuerlein, I., and Spada, H. The active integration of information during learning with dynamic and interactive visualisations. *Learning and Instruction 14*, 3 (2004), 325–341.

[6] Buhrmester, M., Kwang, T., and Gosling, S. D. Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on psychological science 6*, 1 (2011), 3–5.

[7] Chase, W. G., and Simon, H. A. Perception in chess. *Cognitive psychology 4*, 1 (1973), 55–81.

[8] Demuth, H. B., Beale, M. H., De Jess, O., and Hagan, M. T. *Neural network design.* Martin Hagan, 2014.

[9] Eilam, B. *Teaching, learning, and visual literacy: The dual role of visual representation.* Cambridge University Press, 2012.

[10] Gentner, D., and Markman, A. B. Structure mapping in analogy and similarity. *American psychologist 52*, 1 (1997), 45.

[11] Gibson, B. R., Rogers, T. T., Kalish, C., and Zhu, X. What causes category-shifting in human semi-supervised learning? In *CogSci* (2015).

[12] Gibson, E. J. Perceptual learning in development: Some basic concepts. *Ecological Psychology 12*, 4 (2000), 295–302.

[13] Glorot, X., Bordes, A., and Bengio, Y. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (2011), pp. 315–323.

[14] Goldstone, R. L., and Barsalou, L. W. Reuniting perception and conception. *Cognition 65*, 2 (1998), 231–262.

[15] Goldstone, R. L., Medin, D. L., and Schyns, P. G. *Perceptual learning.* Academic Press, 1997.

[16] Goldstone, R. L., Schyns, P. G., and Medin, D. L. Learning to bridge between perception and cognition. *The psychology of learning and motivation 36* (1997), 1–14.

[17] Harel, A. What is special about expertise? visual expertise reveals the interactive nature of real-world object recognition. *Neuropsychologia 83* (2016), 88–99.

[18] Jamieson, K. G., Jain, L., Fernandez, C., Glattard, N. J., and Nowak, R. Next: A system for real-world development, evaluation, and application of active learning. In *Advances in Neural Information Processing Systems* (2015), pp. 2656–2664.

[19] Kellman, P. J., and Massey, C. M. Perceptual learning, cognition, and expertise. *The psychology of learning and motivation 58* (2013), 117–165.

[20] Koedinger, K. R., Corbett, A. T., and Perfetti, C. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive science 36*, 5 (2012), 757–798.

[21] Maas, A. L., Hannun, A. Y., and Ng, A. Y. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml* (2013), vol. 30, p. 3.

[22] Mayer, R. E. *Cognitive theory of multimedia learning.* The cambridge handbook of multimedia learning (2nd ed., pp. 31-48). New York, NY: Cambridge University Press, 2009.

[23] NRC. *Learning to Think Spatially.* Washington, D.C.: National Academies Press, 2006.

[24] Patil, K. R., Zhu, X., Kopeć, Ł., and Love, B. C. Optimal teaching for limited-capacity human learners. In *Advances in neural information processing systems* (2014), pp. 2465–2473.

[25] Peirce, C. S., Hartshorne, C., and Weiss, P. *Collected Papers of Charles Sanders Peirce: (Vol. I-VI).* MA: Harvard University Press, 1935.

[26] Qualtrics. Qualtrics©2018. `https://it.wisc.edu/services/surveys-qualtrics`, last visited 01-18-2018, 2005.

[27] Rau, M. A. Conditions for the effectiveness of multiple visual representations in enhancing stem learning. *Educational Psychology Review 29*, 4 (2017), 717–761.

[28] Rau, M. A., Aleven, V., and Rummel, N. Successful learning with multiple graphical representations and self-explanation prompts. *Journal of Educational Psychology 107*, 1 (2015), 30.

[29] Rau, M. A., Mason, B., and Nowak, R. D. How to model implicit knowledge? similarity learning methods to assess perceptions of visual representations. In *Educational Data Mining* (2016), pp. 199–206.

[30] Rau, M. A., Michaelis, J. E., and Fay, N. Connection making between multiple graphical representations: A multi-methods approach for domain-specific grounding of an intelligent tutoring system for chemistry. *Computers & Education 82* (2015), 460–485.

[31] Richman, H. B., Gobet, F., Staszewski, J. J., and Simon, H. A. Perceptual and memory processes in the acquisition of expert performance: The epam model. *The road to excellence: The acquisition of expert performance in the arts and sciences, sports, and games* (1996), 167–187.

[32] Schnotz, W. An integrated model of text and picture comprehension. *The Cambridge handbook of multimedia learning (2 ed., pp. 72-103)* (2014).

[33] Shanks, D. R. Implicit learning. *Handbook of cognition* (2005), 202–220.

[34] Soderstrom, N. C., and Bjork, R. A. Learning versus performance: An integrative review. *Perspectives on Psychological Science 10*, 2 (2015), 176–199.

[35] Uttal, D. H., and Doherty, K. O. Comprehending and learning from 'visualizations': A developmental perspective. *Visualization: Theory and practice in science education* (2008), 53–72.

[36] Wise, J. A., Kubose, T., Chang, N., Russell, A., and Kellman, P. J. Perceptual learning modules in mathematics and science instruction. *Teaching and learning in a network world'(IOS Press, 2000)* (2003), 169–176.

[37] Wylie, R., and Chi, M. T. The self-explanation principle in multimedia learning. *R. E. Mayer (Ed.), The Cambridge handbook of multimedia learning* (2014), 413–432.

[38] Zhu, X. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *AAAI* (2015), pp. 4083–4087.

[39] Zhu, X., Singla, A., Zilles, S., and Rafferty, A. N. An overview of machine teaching. *arXiv preprint arXiv:1801.05927* (2018).

[40] Ziegler, E., and Stern, E. Delayed benefits of learning elementary algebraic transformations through contrasted comparisons. *Learning and Instruction 33* (2014), 131–146.

Mining User Trajectories in Electronic Text Books

Ahcène Boubekki
Leuphana University of Lüneburg
boubekki@leuphana.de

Shailee Jain[*]
The University of Texas at Austin
shailee@cs.utexas.edu

Ulf Brefeld
Leuphana University of Lüneburg
brefeld@leuphana.de

ABSTRACT

Analyzing user behavior in electronic textbooks offers appealing insights into how pupils interact with the book and internalize the content. Using these insights may help to personalize the book, e.g., to support users with special educational needs. Conventional approaches often focus on atomic, user-triggered events like clicks or scrolls. In this paper, we propose to view all ongoing sessions in a classroom simultaneously and cast the problem as a multi-user problem over space and time. We devise two distance measures to compare the navigation behavior of pupils in different dimensions. Empirically, we observe that our metrics lead to interpretable clusters and serve as performance indicators.

Keywords

Sequential clustering, behavioral analyses, spatio-temporal trajectories

1. INTRODUCTION

The advent of information and communication technologies (ICT) in education has given teachers and educators a magic box full of possibilities [21]. Learning can now be made interactive and engaging for students. The digitization movement has further expanded with MOOCs [18, 10] that provide easy access to extensive and high quality courses online. Situated in-between traditional classrooms and online MOOCs, are electronic textbooks.

E-books incorporate the benefits of both traditionally printed copies and online media. Their structure closely resembles real books, thus rendering a look and feel familiar to students and teachers alike. Additionally, they often include interactive objects (hyperlinks, text boxes for comments) and interlinked media types to enhance the learning experience and delineate content better. Teachers can easily integrate the new technology in their classroom as they offer the full bandwidth, from traditional reading to creative exploring tasks. In addition, electronic books are usually designed to be self-contained and prevent the risk of students being lost in large amounts of content.

This work is part of a project that aims to evaluate the effectiveness of electronic textbooks as learning tools. Our study is based on a collaboration with psychologists and educators. The premise is an electronic text book called the 'mBook' [27, 28] that has been written and developed by a team of history teachers and didacticians. It is being deployed in the German-speaking community of Belgium since 2013.

The mBook records all user-triggered events like clicks and scroll operations such that every session can be replayed entirely. Quantities like the visible content at each timestamp can be derived straight forwardly from this data. We aim to use this information to identify usage patterns in the behavior of the pupils and analyze how they reflect on their performances.

Extracting patterns from log files has been a widely researched topic. Usual techniques range from Behavioral Sequential Analysis [2, 31, 9] to mixtures of Markov chains [6, 7, 15, 8]. However, all these methods are based on event transitions and do not consider historical events or past data. Higher-order Markov chains could possibly handle longer sequences that condition these transitions. Nevertheless, the computation becomes rapidly intractable.

The approach we choose here is to literally extend the navigation metaphor and build a structure to handle sessions as is they were spatio-temporal trajectories. For this purpose, we first extend the shortest path distance in a graph to handle extra events like the loss of focus. Secondly, we build a distance metric to compare trajectories independent of their length and duration. This measure is especially built for our use-case since it not only measures extent of difference between topics studied by two users, but also quantifies the differences in their navigation behavior. Such diverse aspects cannot be fully captured by traditional approaches that rely on simple statistics like the number of pages viewed. Additionally, by comparing navigation patterns between classmates, we characterize teaching style and detect outliers or specific learning patterns.

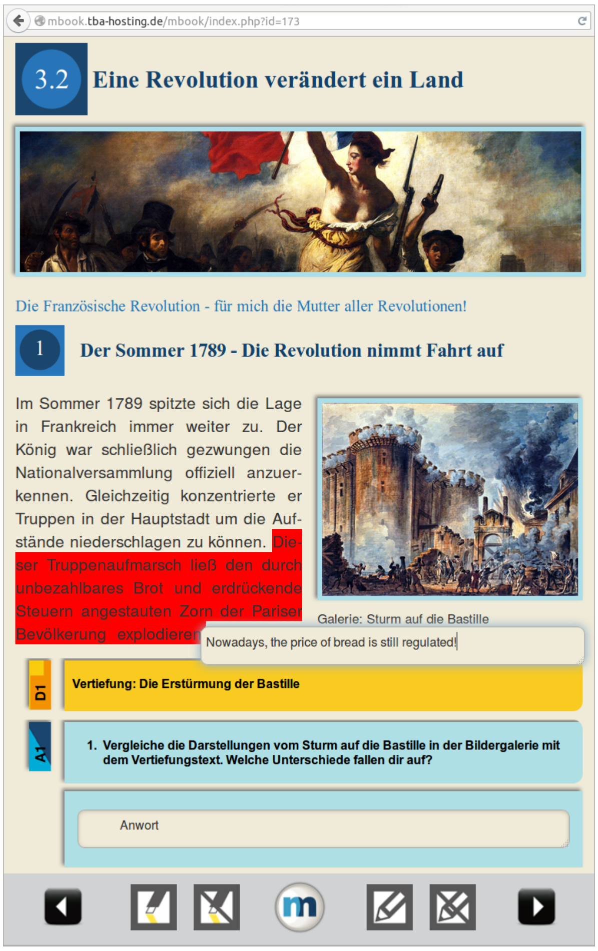
The rest of the paper is structured as follows. In Section

[*]Work done while at National Institute of Technology Karnataka, India.

2, we briefly introduce the mBook project. Notations and concepts necessary for the construction of the distance are presented in Section 3. We also review existing distance metrics based on three properties that a trajectory distance should satisfy, to successfully capture pupils' navigation patterns. Our page and trajectory distances are built in Section 4. In Section 5.1, the clustering qualities of our contribution are highlighted. Finally, in Section 5.2, we study how behavior patterns influence pupils performances and depend on the teaching style.

2. MBOOK

The mBook [27] is an electronic textbook for history, developed for students from grades 6 to 9. It is a part of a project regrouping didacticians, psychologist and computer scientists to study the influence of ICT on pupils and teaching staff. The ebook itself is a website based on a Typo3 environment so that it can be used independently of the device. However, tablets are the predominant device in most classrooms. The primary organization of the book is in the form of web-pages, grouped to represent different chapters/content. The book has 5 chapters that cover Antiquity, Middle Age, Renaissance, 19th Century, and the 20th and 21st Centuries. It also has an additional chapter on methods.



Eine Revolution verändert ein Land

Die Französische Revolution - für mich die Mutter aller Revolutionen!

Der Sommer 1789 - Die Revolution nimmt Fahrt auf

Im Sommer 1789 spitzte sich die Lage in Frankreich immer weiter zu. Der König war schließlich gezwungen die Nationalversammlung offiziell anzuerkennen. Gleichzeitig konzentrierte er Truppen in der Hauptstadt um die Aufstände niederschlagen zu können. Dieser Truppenaufmarsch ließ den durch unbezahlbares Brot und erdrückende Steuern angestauten Zorn der Pariser Bevölkerung explodieren

Galerie: Sturm auf die Bastille

Nowadays, the price of bread is still regulated!

Vertiefung: Die Erstürmung der Bastille

1. Vergleiche die Darstellungen vom Sturm auf die Bastille in der Bildergalerie mit dem Vertiefungstext. Welche Unterschiede fallen dir auf?

Figure 1: Screenshot of the mBook.

Content types cover five main components: text, galleries, audios or videos, information areas and a navigation bar. The primary content is in the form of text. A student can add notes to the text or highlight parts of it. Galleries comprise of pictures related to the text. Some audio or video files are directly integrated to the web-page and can be visualized from there. Information areas below the text provide additional information, beyond what is assigned for the chapter. These are usually organized in boxes that can be opened and accessed with a click/keypress event. Finally, the navigation bar at the bottom of the page allows the student to traverse sections and create highlights or notes. The section traversals include moving to either the previous, current or next section pages. In total, there are 738 pages, including 478 galleries and 537 exercises. Every page is assigned a unique identifier.

Since its deployment, the mBook was used by about 3,000 students in seven schools of the German-speaking community of Belgium. Since 2013, approximately 40,000 sessions were initiated and more than 7 million events (clicks, scrolls, key press, etc.) were tracked.

The project overseeing the deployment of the ebook also organized standardized tests at the end of each academic year. Based on these tests, the competency and knowledge of the pupils in history was regularly assessed using a Rasch model [23]. Additional variables like motivation, IT access and IT skill were obtained by questionnaires and MCQ tests.

3. PRELIMINARIES

In this section, we introduce notation and concepts that will become handy in sections to follow.

3.1 Notations

We begin with formally introducing trajectories.

Definition 1 (Trajectory). *Let Ω be a set. A trajectory $X = (x_i, t_i)_{0 \le i \le N}$ on Ω is a sequence of points x_i of Ω and of time-stamps t_i counted relative to t_0 such that $t_i \le t_{i+1}$. The length of the trajectory X is $N + 1$ and its duration is t_N.*

When the time component is not relevant, the t_i will be omitted. To ease legibility, a sequence $(x_i)_{0 \le i \le N}$ will be abbreviated $(x_i)_N$ whenever the context allows.
Trajectories are essentially time-series of spatial points. In order to later have a notion of similarity between two trajectories, one needs to have a notion of distance between two points. A sequence of elements of Ω is an element of the power set of Ω. Thus, we give an abstract definition of a *distance* that could then be used for points or sequences of points.

Definition 2 (Distance). *Let Ω be a set. The function $d : \Omega \times \Omega \to \mathbb{R}$ is called a* distance *if it satisfies these properties for any elements $x, y, z \in \Omega$:*

- $\Delta(x, x) = 0$,
- *Non-negativity:* $\Delta(x, y) \ge 0$,
- *Symmetry:* $\Delta(x, y) = \Delta(y, x)$.

It is a metric *if it also satisfies:*

- *Identity of indiscernibles:* $\Delta(x, y) = 0 \Leftrightarrow x = y$,
- *Triangle inequality:* $\Delta(x, z) \leq \Delta(x, y) + \Delta(y, z)$.

In the following, we will prefer the notion of *distance* which is less restrictive than a *metric*. However, the distinction can be crucial to some clustering algorithms such as DBSCAN [12, 19] or k-medoids [17, 3] that assume the triangle inequality holds and thus require a metric between points. Other approaches like k-means and many hierarchical clustering methods [24] work well with non-metric distances. One exception is Ward's method [30] that is even more restrictive and relies on Euclidean distance.
Since every metric is also a distance, in the remainder, we denote generic distances between points and trajectories using d and Δ respectively.

3.2 Requirements

The aim of the work is to regroup pupils trajectories of various durations, within the mBook. This grouping should depend on the visited pages and be independent of session start. Additionally, we would like similar behaviors to be regrouped together. This can be controlled by enforcing the distance to satisfy certain properties.

P1: If Y last longer than X, for any truncation Y' of Y lasting longer than X, $\Delta(X, Y') = \Delta(X, Y)$.

P2: If X' and Y' go through the same sequence of points as X and Y but slower (or faster), $\Delta(X, Y) = \Delta(X', Y')$.

P3: If X and Y are loops, i.e. they start and end at the same point, their n-iterations are denoted as X^n and Y^n. If X and Y have the same duration, then $\Delta(X^n, Y^n) = \Delta(X, Y)$.

To motivate these three properties, we will make use of an analogy using a track and field race. Let X and Y be competing athletes and Δ an observer measuring the distance between the runners. Once one of the athletes finishes the race or gives up, the competition ends and Δ cannot make any further measurements. This is what property P1 encloses.
Now suppose that two other competitors X' and Y' perform exactly like the previous ones, but they run at half the speed of X and Y. Δ would make the same observations as above, relative to the total duration of the race. Hence, as stated in P2, we require that $\Delta(X, Y) = \Delta(X', Y')$.
To illustrate P3, X and Y finish the first lap in the same time. They continue similarly for the remaining laps.Thus, the information Δ extracts is the same for every lap. In other words, as stated in P3, $\Delta(X^n, Y^n) = \Delta(X, Y)$.

The first property P1 implies that a trajectory and its sub-trajectories are considered as equal. Sequences of different lengths or durations can then have a distance of 0. Consequently, the identity of indiscernibles is prohibited. Note that property P2 requires that $\Delta(X, Y) = \Delta(X', Y')$, however in the general case, $\Delta(X, Y) \neq \Delta(X, Y')$.

3.3 Distances

Distances on trajectories can be split into two groups [5]: shape-based and warping-based approaches. Warping-based approaches [4, 29] aim at handling sequences of various length by finding an alignment that minimizes a cost function. Dynamic Time Warping (DTW) [4] is often used in speech recognition tasks, but can be leveraged for any type of time series. The main limitation of this measure is that the evaluation algorithm is computationally demanding and has a time complexity of $O(N^2)$ in the length of the longest trajectory. Approximations have been developed to bring the complexity to an almost linear asymptote [26] but at the cost of a lower precision.

Definition 3 (DTW). *Given two trajectories* $X = (x_i)_N$ *and* $Y = (y_j)_M$*, dynamic time warping (DTW) computes an alignment* $W = (w_k)_K$ *with the following properties:*

- $w_k = (x_i, y_j)$, $1 \leq i \leq N$, $1 \leq j \leq M$,
- $w_1 = (x_1, y_1)$,
- $w_K = (x_N, y_M)$,
- $d(w_k) = d(x_i, y_j)$,
- $w_k = (x_i, y_j) \Rightarrow w_{k+1} \in \left\{ \begin{array}{c} (x_i, y_{j+1}) \\ (x_{i+1}, y_j) \\ (x_{i+1}, y_{j+1}) \end{array} \right\}$.

Finally the distance between X *and* Y *is then given by:*

$$\mathrm{DTW}(X, Y) = \min_W \sum_{k=1}^{|W|} d(w_k).$$

The final result is the sum of the distances of the aligned points. Hence, the value grows with the length of the trajectories. This prevents DTW from satisfying P1 and P3. Note that the time-stamps are not considered here. As a consequence, P2 is naturally satisfied given that the duration between two points is irrelevant.

Shape-based distances aim at capturing geometric properties of the trajectories. A representatives of this family are for example Hausdorff [16], as well as more recent ones like the One-Way-Distance [20] and the Symmetrized Segment-Path Distance [5].

Definition 4 (Hausdorff). *Given two trajectories* $X = (x_i)_N$ *and* $Y = (y_j)_M$*. The Hausdorff distance is defined as*

$$\mathrm{HAUS}(X, Y) = \max \left(\sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y) \right).$$

The Hausdorff distance is independent of the timestamps of the points, hence property P2 is satisfied; the computation relies only on their distribution. The number of times each point is visited does however influence the distance. In

particular the situation described by P3 is holds.
A limitation of this measure is that it can be easily deceived by odd point distributions. Consider the three trajectories X, Y and Z represented in Figure 2. Although the shapes are very different, $Haus(X, Z) = Haus(Y, Z) = 3$.
If the last point of X were removed, $Haus(X, Z)$ would decrease. This is in contradiction with P1.

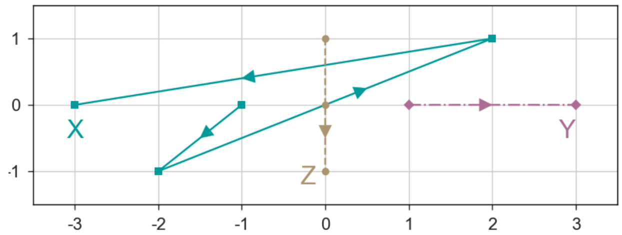


Figure 2: Three trajectories on the plane such that $Haus(X, Z) = Haus(Y, Z) = 3$. The arrows indicate the points orders.

The definitions of the One-Way-Distance (OWD) and Symmetrized Segment-Path Distance (SSPD) require to define the distance from a point to a trajectory:

Definition 5 (Distance Point-Trajectory). *Let x be a point of Ω and $Y = (y_j)_M$ be a trajectory. A segment of Y is a pair of successive points of Y, $[y_j, y_{j+1}]$. The distance between x and a segment of Y is the shortest distance between x and any point of the segment:*

$$d(x, [y_j, y_j + 1] = \min_{\tau \in [0,1]} (d(x, y_j\tau + (1-\tau)y_{j+1}))$$

The distance between x and Y is the shortest distance between x and the segments of Y:

$$d(x, Y) = \min_j d(x, [y_j, y_j + 1]).$$

Definition 6 (OWD). *The one-way-distance (or OWD) between two trajectories $X = (x_i, t_i)_N$ and $Y = (y_j, t'_j)_M$ is defined as the integral of the distance from points of X to trajectory Y divided by the duration of X :*

$$OWD(X; Y) = \frac{1}{t_N} \int_{x \in X} d(x, Y)dx.$$

The symmetric OWD is the average of the OWD between X and Y:

$$sOWD(X, Y) = \frac{OWD(X; Y) + OWD(Y; X)}{2}.$$

The sOWD is close to the distance we want to build. Thanks to the normalization with duration, the measure satisfies P2 and P3. However it is not invariant per truncation as required by P1. If Y is truncated into Y', the duration of the later is shorter than the former, hence $OWD(Y'; X) \neq OWD(Y; X)$ in general.
Given that Y' is said in P1 to last longer than X, $OWD(X; Y') = OWD(X; Y)$. Yet, $\frac{1}{2}(OWD(X; Y') + OWD(Y'; X))$ is different from $\frac{1}{2}(OWD(X; Y) + OWD(Y; X))$ in general.

Definition 7 (SSPD). *The Segment-Path Distance, SPD, between two trajectories $X = (x_i)_N$ and $Y = (y_j)_M$ is :*

$$SPD(X; Y) = \frac{1}{N+1} \sum_{i=0}^{N} d(x_i, Y).$$

The Symmetric Segment-Path Distance is the average of the SPD between X and Y:

$$SSPD(X, Y) = \frac{SPD(X; Y) + SPD(Y; X)}{2}.$$

The distance SSPD is independent of the time indexing, hence P2 is automatically satisfied. Besides the normalization by the number of points assure that the distance between loop trajectories is invariant with the number of iterations. Thus SSPD complies with P3.
However similarly than for OWD, the Symmetric Segment-Path Distance does not satisfy P1. Indeed if Y last longer than X and Y' is a truncation Y lasting as well longer than X, $SPD(Y'; X) \neq SPD(Y; X)$ while $SPD(X; Y') = SPD(X; Y)$. The averages are hence also different.

4. WEB TRAJECTORIES

Consider a website $\mathcal{W}$ whose structure is given by the page graph $G = (\mathcal{P}, \mathcal{E})$. We refer to the corresponding web-page of a node $p \in \mathcal{P}$ by $\mathcal{W}(p)$. That is, a node $p \in \mathcal{P}$ has a child $p' \in \mathcal{P}$ if users can transfer from page $\mathcal{W}(p)$ to $\mathcal{W}(p')$ by clicking a link or using the navigation bar. In that case $(p, p') \in \mathcal{E}$ holds. A loss of focus happens when the user turns off the screen of the tablet, or visit another tab. In order to handle this event, we add a dummy page F to $\mathcal{P}$. As it can happen anytime, F is connected to all the other pages.

A session on $\mathcal{W}$ can be represented as a sequence of pairs $P = (p_i, t_i)_{0 \leq i < l}$, where a user views page $\mathcal{W}(p_i)$ at timestamp t_i. For simplicity, we represent timestamps relatively to t_0, to retain the elapsed time on page and site. To call P a trajectory, we need to define a metric between its points.

4.1 Distances between pages

A natural distance measure for pages is the shortest path between the corresponding nodes in the underlying graph G. However, the auxiliary state F needs to be appropriately incorporated to allow for a meaningful application of a shortest path algorithm. Despite being connected to all the pages, we thus set the distance between F and any other page p to $dF \in \mathbb{R}_+$ such that

$$\max_{p,q \in \mathcal{P}} \textsc{ShortestPath}(p, q) < dF.$$

We motivate this choice by the fact that we want the clustering algorithm to consider a loss of focus as a special state. By making it very costly with respect to the other costs, we favor clusters of sessions that frequently visit F.

Definition 8 (Page Distance). *The distance d between two pages $p, q \in \mathcal{P}$ is defined* *as follows.*

$$d\,(p,q) = \begin{cases} \textsc{ShortestPath}(p,q) & ,\ \textit{if}\ p \neq F\ \textit{and}\, q \neq F \\ dF & ,\ \textit{if}\ p \neq F\ \textit{and}\, q = F \\ 0 & ,\ \textit{if}\ p = F\ \textit{and}\, q = F \end{cases}$$

This *page distance* now allows the comparison of points inside a page graph and can be used by existing measures comparing trajectories. In order to assure that its usage does not remove the distance properties out of these measures, d needs to be a distance as well.

Lemma 1. *The functions $d : \mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R}$ is a metric.*

Proof. Non-negativity, symmetry and the identity of indiscernibles directly apply from the ShortestPath which is a metric on $\mathcal{P} \setminus F$.
Let us prove the triangle inequality, i.e for p, q, s in $\mathcal{P}$: $d(p, r) \leq d(p, q) + d(q, r)$

- If $r = F$ and $q = F$, $d(F, F) = 0$.
- If $r = F$ and $q \neq F$, per non-negativity of d: $d(p, F) \leq dF \leq d(p, q) + dF = d(p, q) + d(q, r)$
- If none of the pages is F, then d is simply the ShortestPath, which satisfies the triangle inequality.

□

4.2 Distances between trajectories

Following Definition 1, sessions can now be viewed as trajectories, more precisely web trajectories. In opposition to spatial trajectories, the position of a web trajectory between two timestamps does not evolve. Hence the position at any timestamp is precisely the one of the most recent point. We define the cross-product C of two trajectories X and Y to keep track the positions changes of X and Y.

Definition 9 (Cross-product). *Let $X = (x_i, t_i)_N$ and $Y = (y_j, t'_j)_M)$ be two trajectories such that $t_N \leq t_M$. The cross-product of X and Y is the sequence $C = C(X, Y) = (c_k)_K = (\bar{t}_k, \bar{x}_k, \bar{y}_k)_{0 \leq k \leq K}$ defined as follows:*

- $\bar{t}_k \in \{t_i, 0 \leq i < N\} \cup \{t'_j, 0 \leq j < M\ \textit{and}\, t'_j \leq t_N\}$
- $c_0 = (0, x_0, y_0)$,
- *For* $0 \leq k < K + 1$, $c_k = (\bar{t}_k, \bar{x}_k, \bar{y}_k)$, *with* $\bar{x}_k = x_i$ *such that* $t_i \leq \bar{t}_k < t_{i+1}$, *and* $\bar{y}_k = y_j$ *such that* $t'_j \leq \bar{t}_k < t'_{j+1}$,
- $c_K = (t_N, x_N, y_j)$ *such that* $t'_j \leq \bar{t}_N < t'_{j+1}$.

Now we devise a distance Δ for web-trajectories. Δ is defined as the normalized area spanned between them until the shortest one ends.

Definition 10 (Trajectory Distance). *Let $X = (x_i, t_i)_N$, $Y = (y_j, t'_j)_M)$ be two trajectories and $C = (\bar{t}_k, \bar{x}_k, \bar{y}_k)_K$ their cross product:*

$$\Delta(X, Y) = \frac{1}{t_N} \sum_{k=1}^{K} d(\bar{x}_{k-1}, \bar{y}_{k-1})(\bar{t}_{k+1} - \bar{t}_k)$$

In Section 3, we formulated three requirements for trajectory distances to assure certain properties in the clustering. The fact that none of the reviewed distances fulfills all of them, motivated the construction of Δ. We will now prove that our distance complies with the three conditions.

Lemma 2. *The function Δ defined on pairs of web-trajectories satisfies the three properties P1, P2 and P3.*

Proof. Let $X = (x_i, t_i)_N$ and $Y = (y_j, t'_j)_M$ be two trajectories and $C = (\bar{t}_k, \bar{x}_k, \bar{y}_k)_{0 \leq k \leq K}$ their cross product. We suppose that Y last longer: $t_N \leq t'_M$. Let us prove that each property is satisfied.

P1: The distance Δ depends only on the cross product of the two trajectories. Per construction, the cross-product contains only the points happening before that the shortest one ends, here X.
Hence for any truncation $Y' = (y_j, t'_j)_{0 \leq j < M'+1}$ of Y such that $M' < M$ and $t_N \leq t'_{M'}$, $C(X', Y) = C(X, Y)$. This implies $\Delta(X, Y') = \Delta(X, Y)$.

P2: For $\lambda > 1$, X' and Y' travel the same path than X and Y but λ times slower means that $X' = (x_i, \lambda t_i)_N$ and $Y' = (y_j, \lambda t'_j)_M)$. Their cross product is $C' = (\lambda \bar{t}_k, \bar{x}_k, \bar{y}_k)_{0 \leq k < K+1}$.

$$\begin{aligned} \Delta(X', Y') &= \frac{1}{\lambda t_N} \textstyle\sum_{k=1}^{K} d(\bar{x}_{k-1}, \bar{y}_{k-1})(\lambda \bar{t}_{k+1} - \lambda \bar{t}_k) \\ &= \frac{\lambda}{\lambda t_N} \textstyle\sum_{k=1}^{K} d(\bar{x}_{k-1}, \bar{y}_{k-1})(\bar{t}_{k+1} - \bar{t}_k) \\ \Delta(X', Y') &= \Delta(X, Y) \end{aligned}$$

P3: We will prove this property for $n = 2$, but it can be extended for any value. In this case X is a loop, i.e. $x_0 = x_N$, and $t_N = t'_M$. A trajectory X^2 traveling two times through X is of duration $2t_N$ and does not visit twice the initial position, i.e.

$$X^2 = (x_i, t_i)_{0 \leq i \leq N} \cup (x_i, t_i + t_N)_{1 \leq i \leq N}.$$

In turn, $C(X^2, Y^2) = (\bar{t}_k, \bar{x}_k, \bar{y}_k)_K \cup (\bar{t}_k + \bar{t}_K, \bar{x}_k, \bar{y}_k)_{1 \leq k \leq K}$. Hence:

$$\begin{aligned} \Delta(X^2, Y^2) = \frac{1}{2t_N} \Big(& \textstyle\sum_{k=0}^{K} d(\bar{x}_{k-1}, \bar{y}_{k-1})(\bar{t}_{k+1} - \bar{t}_k) \\ & + d(\bar{x}_K, \bar{y}_K)(\bar{t}_K + (\bar{t}_1 + t_N)) \\ & + \textstyle\sum_{k=1}^{K} d(\bar{x}_{k-1}, \bar{y}_{k-1})((\bar{t}_{k+1} + t_N) - (\bar{t}_k + t_N)) \Big) \end{aligned}$$

Given that $t_N = t'_M$ and that X and Y are loops, $\bar{x}_K = x_N = x_0$, $\bar{y}_K = y_N = y_0$ and $\bar{t}_K = t_N$. Besides following Definition 9 $\bar{t}_0 = 0$. Consequently ,

$$d(\bar{x}_K, \bar{y}_K)(\bar{t}_K + (\bar{t}_1 + t_N)) = d(\bar{x}_0, \bar{y}_0)(\bar{t}_0 + \bar{t}_1)$$

. This term can hence be integrated inside the second sum, such that we have:

$$\begin{aligned}\Delta(X^2,Y^2) &= \frac{1}{2t_N}\left(\sum_{k=0}^{K} d(\bar{x}_{k-1},\bar{y}_{k-1})(\bar{t}_{k+1}-\bar{t}_k)\right.\\ &\quad \left.+\sum_{k=0}^{K} d(\bar{x}_{k-1},\bar{y}_{k-1})(\bar{t}_{k+1}-\bar{t}_k)\right)\\ &= \frac{1}{2t_N}\left(2\sum_{k=0}^{K} d(\bar{x}_{k-1},\bar{y}_{k-1})(\bar{t}_{k+1}-\bar{t}_k)\right)\\ \Delta(X^2,Y^2) &= \Delta(X,Y)\end{aligned}$$

□

Algorithm 1: $\Delta(X,Y)$

```
$\Delta \leftarrow 0$;
$T \leftarrow \min(t_N, t'_M)$;
Initialize a list $C$ with $(0, x_0, y_0)$
foreach $(t_i, x_i)$ in $X$ with $i > 0$ and $t_i \leq T$ do
    Append $(t_i, x_i, NAN)$ to $C$;
end
foreach $(t'_j, y_j)$ in $Y$ with $j > 0$ and $t'_j \leq T$ do
    Append $(t'_j, NAN, y_j)$ to $C$;
end
Sort $C$ accordingly to the first column;
$K \leftarrow$ length of $C$;
for $1 \leq k \leq K$ do
    $C_{k-1} = (t_{k-1}, x_{k_1}, y_{k-1})$;
    $C_k = (t_k, x_k, y_k)$;
    $\Delta \leftarrow \Delta + d(x_{k-1}, y_{k-1})(t_k - t_{k-1})$
    if $x_k$ is $NAN$ then
        $x_k \leftarrow x_{k-1}$;
    end
    if $y_k$ is $NAN$ then
        $y_k \leftarrow y_{k-1}$;
    end
end
Return $\Delta/T$;
```

Algorithm 1 describes an efficient way to compute Δ. Firstly, the distance Δ initialized to 0 and the shortest duration T is retrieved. The cross product C is a list of triplets : (t_k, x_k, y_k). The first coordinate indicates the timestamps, the two others the positions of X and Y at this time. The first tuple gives the initial positions of the two trajectories. Then all the positions of X and Y with a timestamp smaller or equal than T are included in C where the position of Y or X is set respectively as unknown. After that C is sorted accordingly to the timestamps.

Finally C is browsed starting from the second element ; Δ is updated accordingly to Definition 10 ; the missing positions are assigned using the last known positions.

Note that if X and Y have points with the same timestamp, C will contains tuples with the same timestamp. It is not problematic as they will cancel out each other during the update of Δ.

The time complexity of Algorithm 1 is $\mathcal{O}(N+M)$. It derives its efficiency from the fact that the assignments of the missing positions in C and the updates of Δ are done in the same loop.

4.3 Example

This section gives an example for the computation of the distance measure Δ. Consider the graph that is displayed in Figure 3. On the left, two trajectories are represented on the page graph. Arrows represent a click that causes a page change. After vising page C, P loses the focus during one time unit. On the right, the progression of the trajectories over time is represented. The x-axis represents time and the y-axis the pages. The distance between P and Q is computed as follows.

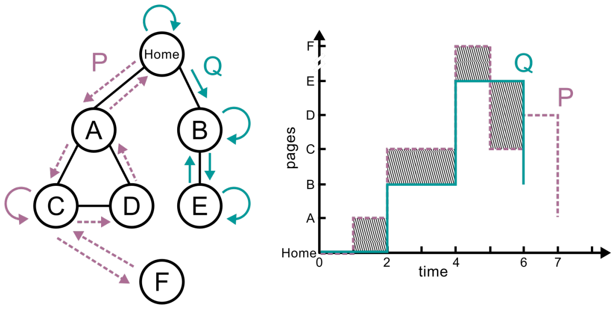


Figure 3: Trajectories on the page graph (left) and as timeseries (right). Edges between F and the other pages are not shown for legibility.

$$\begin{aligned}\Delta(P,Q) &= \tfrac{1}{6}\left[d(H,H)+d(A,H)+d(C,B)*2\right.\\ &\qquad\left.+d(F,E)+d(C,E)\right]\\ \Delta(P,Q) &= \tfrac{1}{6}\left[0+1+3*2+dF+4\right]\\ \Delta(P,Q) &= \frac{11+dF}{6}\end{aligned}$$

5. EMPIRICAL RESULTS

5.1 Clustering

In this section, we report on clustering results that are obtained by using Hausdorff, DTW and the proposed Δ distances. We use K-means [24] as the underlying clustering algorithm. The distance of a trajectory to a cluster is the average distance between the trajectory and all the sessions in the cluster. We repeat every experiment 50 times and report on the best result for every measure.

The requirements stated in Section 3 aim to promote groupings of sessions that share long subsequences of viewed pages. To highlight the consequences of these choice, we restrict the data to only a single day. The subset contains 41 sessions from 37 users with an average duration of 32 minutes. The small scale allows for an interpretable analysis of the resulting clusterings. However, note that the computational complexity of DTW and Hausdorff quickly become infeasible with more data: The computation of the upper triangle of the DTW distance matrices using [4] requires more than 6 hours.

Although the sessions do not contain information about teachers, we will still evaluate the clusterings based on their similarity with the teachers' groupings. They should not be very different. Indeed, during one class, pupils tend to worked on the same subject. Thus, we expect them to be clustered together.

The teacher ID of the pupils behind session are represented by the y-axis of Figure 4.a. The connection times (x-axis) show six different classes. An analysis of the session logs shows that the closest classes in terms of topic and thus also

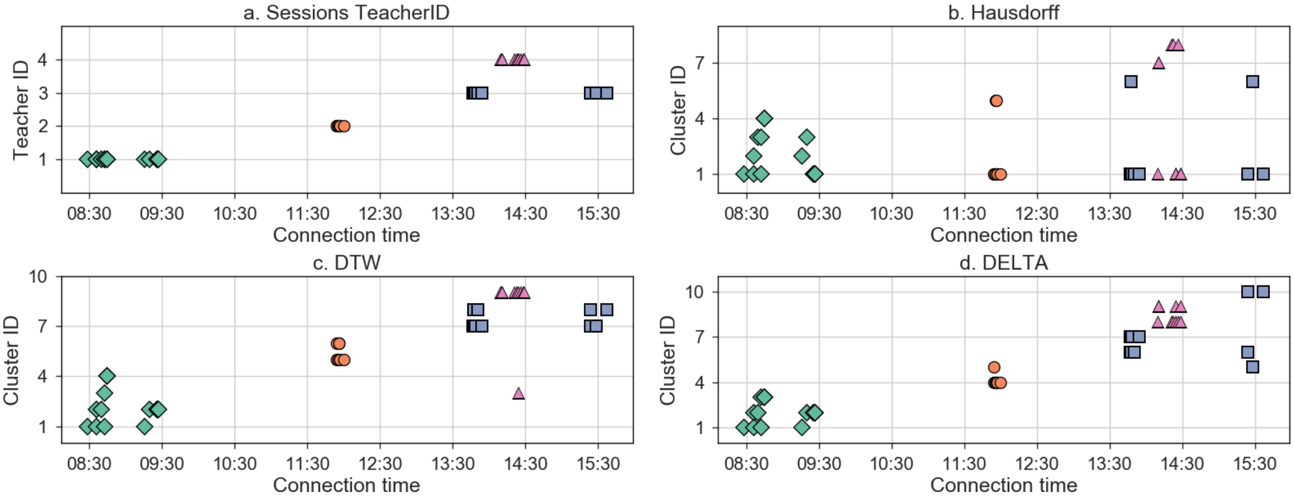


Figure 4: Teacher and cluster assignments of each sessions.

in terms of distance in the web-site graph are the ones of teacher 1 and 3, who dedicated all their lessons of this day to Alexander the Great and to the Roman Empire respectively. During a single class, teacher 2 focused on the situation of Belgium during WWII. The group of teacher 4 learned about the Reformation.

Two settings are evaluated. In the first one the number of clusters K is fixed to the number of teachers, that is $K = 4$. In the second experiment, K is chosen an order of magnitude higher to give the algorithm enough degrees of freedom to return the optimal amount of clusters for every measure. The returned clusters in this last setting are plotted in Figures 4.b to d. The final number of clusters found by each method and the homogeneity scores [25] of the clustering relatively to the teachers' distribution are given in Table 1. A homogeneity score of 1 indicates that no cluster contains sessions from multiple teachers.

Table 1: Number of clusters and homogeneities in the case of constrained or unconstrained clusterings.

	K=4		K=20	
Distance	# Cl.	Homog.	# Cl.	Homog.
Hausdorff	4	0.14	8	0.39
DTW	4	0.67	9	0.97
Δ	4	0.87	10	0.97

In both settings, the Hausdorff distance performs poorly. As shown in Figure 4.b, it fails at detecting class behaviors. The first cluster is spread all over the day, despite that each class studied different sections. By contrast, Δ's high homogeneities indicates that our proposed distance successfully detects the topics. Even when K is fixed to 4, Δ outperforms DTW and made few clustering errors. For $K = 20$, DTW and Δ create enough clusters such that all of them are pure with respect to the teacher, except for one session that is wrongly assigned in a cluster with sessions from another teacher. Interestingly for both distances, this mistake happens in a group of two sessions. DTW groups two sessions from teacher 1 and teacher 4 together, while Δ mistakenly associates a session from teacher 2 with a session from teacher 3, respectively.

For $K = 20$, the main difference between DTW and Δ is how they handle teacher 4. While DTW aims to group sessions associated with teacher 4 together, our distance measure splits them into two clusters. The trajectories of each cluster for each measure are shown in Figure 5. The pages are organized per chapter.

DTW detects the topic well as all the sessions dealing with Renaissance are grouped together. Cluster 3 in Figure 5.a is actually the DTW's cluster that is made only of two sessions from two different teachers. It is not clear why this artifact occurs. By contrast, our distance measure creates two groups out of all trajectories visiting the Renaissance's chapter. Cluster 8 shown in Figure 5.c contains those sessions that navigate more or less directly to the page about the Reformation and then stay on that page until the session is terminated. Sessions with more irregular trajectories are put into cluster 9. Thus, in addition to the topic, the shape of the trajectories is also a determining factor for Δ-based clusterings.

This section showed that pupils may exhibit very different types of behavior during the same class and that our distance measure performs well in detecting these behaviors. The next section investigates how the behaviors relate to the pupils performance in the class.

5.2 Assessments

In this section, we study the relation between the expressed behavior and the pupil's scores described in Section 2.

The activity of a user during one session can be measured through statistics like the 'number of pages seen per minute' (PPM) or the 'number of events per minute' (EPM). The average distance between a pupil's session and the other class sessions indicates how much the pupil's usage diverges from the group's.
However, these values can not be used to compare the activity between classes. Indeed, in a class with an average

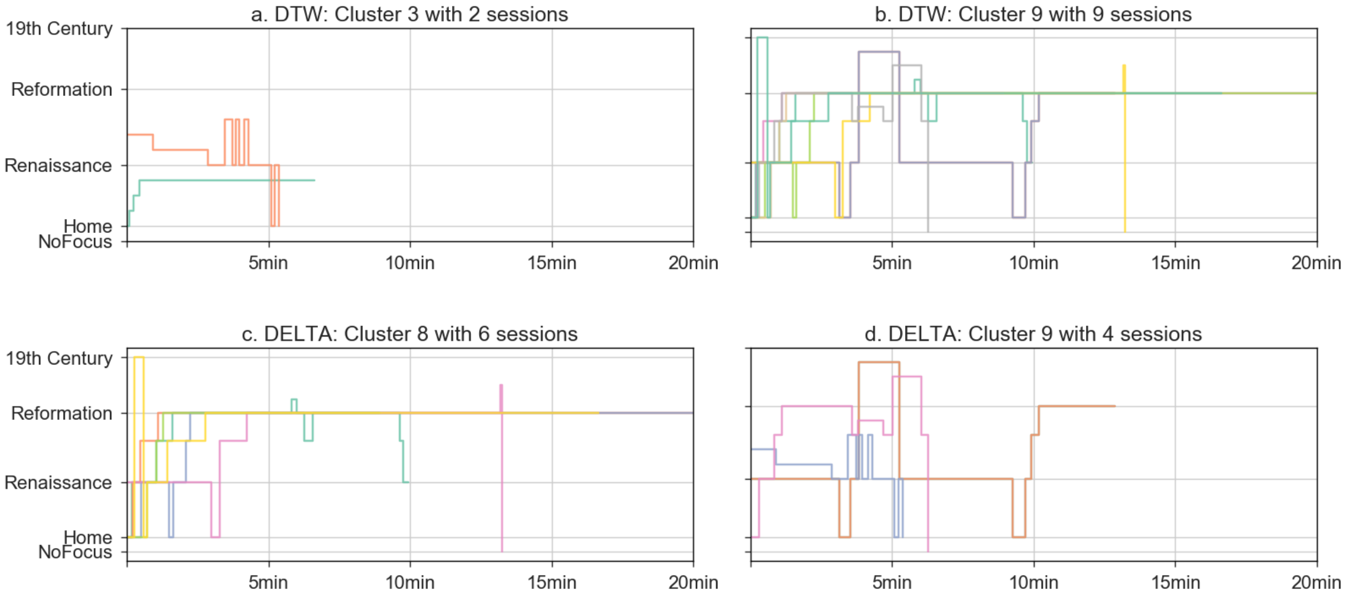


Figure 5: Trajectories of clusters obtained using DTW and Δ associated to the class of teacher 4.

of one page view per minute, a user viewing one page per minute will be considered as regular. However if the average of the class was 3, the same user would appear too inactive. Hence, these quantities need to be expressed relative to the average value of each class.

The average distance between trajectories of a class, also called the intra-class distance, is denoted as Ψ. The average distance of session P to the other class trajectories, also called divergence of the session, is denoted as $\psi(P)$.

We extract 400 class-sessions between February and July 2017, under the supervision of two teachers in two different schools. A class-session happens between 08:00 and 16:00 and contains at least five sessions from pupils with the same teacher that all start within 10 minutes. Table 3 contains the number of classes, sessions associated to the teacher, as well as the number of pupils. The average intra-class distance of the teachers' classes are given in the last column with standard deviations. Correlations between the measures and the pupils' scores are reported in Table 2. Pearson's correlations with a p-value smaller than 5% are marked in bold face. The displayed numbers indicate that the two groups show different behavior and that the teachers apply different teaching styles.

Table 2 suggests that while the three indicators correlate with the pupils competencies, they do so in different directions. For instance, pupils that possess a higher ψ, visit more pages per minute or interact more than the other pupils, during the same class. These pupils of teacher A perform better at the competency test. The opposite holds for the pupils of teacher B.

These differences can be interpreted only if put in the context of the average intra-class distances, given in Table 3. A Mann-Whitney U test [22, 13] between the Ψ of the two teachers' classes returns a U-value of 85 ($<$ 87 critical) and a one-sided p-value of 0.02. Thus, we can state that the pupils in teacher B's classes have more definite trajectories. And pupils who diverge from the predominant path tend to perform worst. To the contrary, the worst performing pupils of teacher A, whose classes present in average a bigger Ψ, are those that under-use the textbook.

The fact that all the indicators correlate with competency could mistakenly be interpreted as redundancy. However, we observe cases where only ψ is significant. For example, a small ψ correlates with high motivation in group A. This is remarkable, since it presents a correlation in the opposite direction of competency.

In the case of teacher B, pupils with low ψ perform better at the competency tests but also possess higher skills in information and communication technologies compared to their classmates. Indeed, among teacher B's pupils, the Pearson coefficient between these two scores indicate a correlation (0.399, p-value 0.0002); PPM and EPM fail to capture this effect.

In addition to the classical PPM and EMP, ψ appears to be a good indicator of the pupils' performances. Besides, it captures relations that are hidden to PPM and EMP and that are independent of connections between different scores.

6. DISCUSSION

In this paper, we focus on methods to extract diverse usage patterns of an e-book, through analysis of spatio-temporal, web-log trajectories. While conventional methods focus on individual events like page-clicks or scrolls, we extract and analyze trajectories within a web-page as a whole. To achieve this, we propose to embed the structure of electronic textbooks into graphs. Once pages of the ebook are associated with nodes in the graph, shortest path algorithms can be applied to compute distances between pages. Additionally, we also lift these distances to entire sessions, by making use of cross-products. The establishment of the distance metrics facilitates the use of spatial clustering methods to sessions of possibly unequal

Table 2: Pearson's correlations and associated p-values for each combination of pupil's activity indicators and score.

Teacher A

	Competency		Knowledge		Motivation		IT Access		IT Skill	
	r	p-value	r	p-value	r	p-value	r	p-value	r	p-value
ψ	**0.179**	0.012	0.096	0.182	**-0.17**	0.017	0.023	0.745	0.092	0.202
PPM	**0.145**	0.044	0.133	0.064	0.039	0.587	-0.002	0.979	0.019	0.789
EPM	**0.185**	0.009	**0.156**	0.03	-0.065	0.37	-0.022	0.761	0.063	0.381

Teacher B

	Competency		Knowledge		Motivation		IT Access		IT Skill	
	r	p-value	r	p-value	r	p-value	r	p-value	r	p-value
ψ	**-0.224**	0.047	-0.165	0.146	0.096	0.402	-0.069	0.547	**-0.357**	0.001
PPM	**-0.232**	0.039	0.049	0.671	0.111	0.331	0.188	0.097	-0.156	0.171
EPM	**-0.232**	0.04	-0.141	0.216	-0.142	0.212	0.081	0.481	0.059	0.604

Table 3: Summary of the analyzed classes.

	#Class	#Sessions	#Pupils	Ψ
Teacher A	27	200	48	5.76 (1.41)
Teacher B	11	80	22	4.48 (1.61)

length.

Empirically, we show that pupils exhibit very different types of behavior during the same class; the proposed distance measure outperforms baseline measures in grouping and detecting these behaviors. Moreover, in another experiment, we show that our distance measure differentiates between teaching styles and facilitates comparison between user behavior and user competence. The average dissimilarity between sessions during a class can thus be turned into an effective indicator of pupil performance and teaching technique. This study thus facilitates a thorough understanding of the effectiveness of e-books, in a classroom setup.

The empirical success of the proposed distance metric establishes it as a useful tool to analyze learning and teaching behaviour in a classroom. We thus hope to further extend these experiments to detect more complex learning patterns, now that a suitable comparison metric has been developed. For instance, our technique could be extended to detect 'outliers' or pupils who completely contravene typical classroom behaviour. It will further be interesting to establish correlations between outliers and performance. This will throw more light on the effectiveness of the teaching style and the ebook medium.

Acknowledgments

This research has been funded in parts by the German Federal Ministry of Education and Science BMBF under grant QQM/01LSA1503C.

7. REFERENCES

[1] H. Alt and M. Godau. Computing the fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5(01n02):75–91, 1995.

[2] R. Bakeman, B. F. Robinson, and V. Quera. Testing sequential association: Estimating exact p values using sampled permutations. *Psychological methods*, 1(1):4, 1996.

[3] S. Baraty, D. A. Simovici, and C. Zara. The impact of triangular inequality violations on medoid-based clustering. In *International Symposium on Methodologies for Intelligent Systems*, pages 280–289. Springer, 2011.

[4] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

[5] P. Besse, B. Guillouet, J.-M. Loubes, and R. François. Review and perspective for distance based trajectory clustering. *arXiv preprint arXiv:1508.04904*, 2015.

[6] A. Boubekki, U. Kröhne, F. Goldhammer, W. Schreiber, and U. Brefeld. Toward data-driven analyses of electronic text books. In *EDM*, pages 592–593, 2015.

[7] A. Boubekki, U. Kröhne, F. Goldhammer, W. Schreiber, and U. Brefeld. Data-driven analyses of electronic text books. In *Solving Large Scale Learning Tasks. Challenges and Algorithms*, pages 362–376. Springer, 2016.

[8] I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White. Visualization of navigation patterns on a web site using model-based clustering. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 280–284. ACM, 2000.

[9] K.-E. Chang, C.-T. Chang, H.-T. Hou, Y.-T. Sung, H.-L. Chao, and C.-M. Lee. Development and behavioral pattern analysis of a mobile guide system with augmented reality for painting appreciation instruction in an art museum. *Computers & Education*, 71:185–197, 2014.

[10] G. Conole. Moocs as disruptive technologies: strategies for enhancing the learner experience and

quality of moocs. *RED. Revista de Educación a Distancia*, (50), 2016.

[11] T. Eiter and H. Mannila. Computing discrete fréchet distance. Technical report, Citeseer, 1994.

[12] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[13] M. P. Fay and M. A. Proschan. Wilcoxon-mann-whitney or t-test? on assumptions for hypothesis tests and multiple interpretations of decision rules. *Statistics surveys*, 4:1, 2010.

[14] M. M. Fréchet. Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, 22(1):1–72, 1906.

[15] C. Geigle and C. Zhai. Modeling student behavior with two-layer hidden markov models. *JEDM| Journal of Educational Data Mining*, 9(1):1–24, 2017.

[16] F. Hausdorff. *Mengenlehre*. Walter de Gruyter Berlin, 1927.

[17] L. Kaufmann and P. Rousseeuw. Clustering by means of medoids. pages 405–416, 01 1987.

[18] R. Kop. The challenges to connectivist learning on open online networks: Learning experiences during a massive open online course. *The International Review of Research in Open and Distributed Learning*, 12(3):19–38, 2011.

[19] M. Kryszkiewicz and P. Lasek. Ti-dbscan: Clustering with dbscan by means of the triangle inequality. In *International Conference on Rough Sets and Current Trends in Computing*, pages 60–69. Springer, 2010.

[20] B. Lin and J. Su. Shapes based trajectory queries for moving objects. In *Proceedings of the 13th annual ACM international workshop on Geographic information systems*, pages 21–30. ACM, 2005.

[21] S. Livingstone. Critical reflections on the benefits of ict in education. *Oxford review of education*, 38(1):9–24, 2012.

[22] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.

[23] G. Rasch. Studies in mathematical psychology: I. probabilistic models for some intelligence and attainment tests. 1960.

[24] L. Rokach and O. Maimon. Clustering methods. In *Data mining and knowledge discovery handbook*, pages 321–352. Springer, 2005.

[25] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007.

[26] S. Salvador and P. Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[27] W. Schreiber, A. Schöner, and F. Sochatzy. *Analyse von Schulbüchern als Grundlage empirischer Geschichtsdidaktik*. Kohlhammer Verlag, 2013.

[28] W. Schreiber, F. Sochatzy, and M. Ventzke. Auf dem weg zu digital-multimedialen lehr-und lernmitteln für kompetenzorientiertes inklusives unterrichten und lernen. *Online Publikation, Medienberatung NRW. Zugriff am*, 11:2017, 2014.

[29] M. Vlachos, G. Kollios, and D. Gunopulos. Discovering similar multidimensional trajectories. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 673–684. IEEE, 2002.

[30] J. H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[31] C. Yin, N. Uosaki, H.-C. Chu, G.-J. Hwang, J.-J. Hwang, I. Hatono, E. Kumamoto, and Y. Tabata. Learning behavioral pattern analysis based on students' logs in reading digital books. 12 2017.

Studying Affect Dynamics and Chronometry Using Sensor-Free Detectors

Anthony F. Botelho
Worcester Polytechnic Institute
100 Institute Rd
Worcester, MA
abotelho@wpi.edu

Ryan S. Baker
University of Pennsylvania
3700 Walnut St
Philadelphia, PA
ryanshaunbaker@gmail.com

Jaclyn Ocumpaugh
University of Pennsylvania
3700 Walnut St
Philadelphia, PA
jlocumpaugh@gmail.com

Neil T. Heffernan
Worcester Polytechnic Institute
100 Institute Rd
Worcester, MA
nth@wpi.edu

ABSTRACT

Student affect has been found to correlate with short- and long-term learning outcomes, including college attendance as well as interest and involvement in Science, Technology, Engineering, and Mathematics (STEM) careers. However, there still remain significant questions about the processes by which affect shifts and develops during the learning process. Much of this research can be split into affect dynamics, the study of the temporal transitions between affective states, and affective chronometry, the study of how an affect state emerges and dissipates over time. Thus far, these affective processes have been primarily studied using field observations, sensors, or student self-report measures; however, these approaches can be coarse, and obtaining finer-grained data produces challenges to data fidelity. Recent developments in sensor-free detectors of student affect, utilizing only the data from student interactions with a computer-based learning platform, open an opportunity to study affect dynamics and chronometry at moment-to-moment levels of granularity. This work presents a novel approach, applying sensor-free detectors to study these two prominent problems in affective research.

Keywords

Student Affect, Affect Dynamics, Affect Chronometry, Deep Learning, Sensor-Free Detectors

1. INTRODUCTION

The various affective states experienced by students during learning have received significant attention from the research community for their prominence in the learning process. Student affect has been shown to correlate with several measures of student achievement [6][22][28], has been found to be predictive of whether students attend college several years later [24], and also whether students choose to take steps towards careers in Science, Technology, Engineering, and Mathematics (STEM) fields [30]. While significant steps have been taken toward understanding the interrelationships between of affect and learning, there are many questions that remain unanswered with regard to how affect is exhibited by students over time as well has how such temporal trends may be informative of student learning outcomes.

The temporality of student affect has been characterized into two areas of study, affect dynamics [31] and affective chronometry. Affect dynamics studies temporal shifts in affect to understand which transitions between affective states are most common. A theoretically-grounded model of affective dynamics has been proposed by D'Mello and Graesser [10], which suggests a typical resolution cycle, where students transition from engaged concentration to surprise to confusion and back to engaged concentration, but which also hypothesizes alternative transitions, including a path from confusion to frustration and boredom.

Affective chronometry also uses temporal measures, but focuses more closely upon how individual affective states (e.g., boredom) behave over time. This was first studied as a special case of affective dynamics, where researchers investigated how frequent it was for an affective state to transition to itself (aka "self-transitions"). More recently, D'Mello and Graesser [9] proposed instead investigating an affective state's "half life," or the decay in the probability of an affective state persisting for a specific duration of time. [9] found evidence that six affective states exhibit exponential decay in their probability over time. That is, the probability that a student remains in a particular state decreases exponentially as the amount of time that the student persists in that state increases. However, engaged concentration (referred to as flow) showed a much slower decay rate than other affective states (e.g., frustration).

There is now a growing body of research in affective dynamics and affective chronometry, commonly using field observations [26][13], or self-reports accompanied by video data [3][9]. These important studies have helped to advance the field, but each method imposes different kinds of limitations on the grain-size of the data. Continuous observation is impractical both for self-report and field observation studies, and it is highly time-consuming for video recording (which can also break down when the student moves away from his or her desk, either for off-task reasons or for on-task purposes like peer-tutoring or requesting assistance). Despite the limitations of these methods, they have often been preferred to sensor-free detectors of affect due to higher reliability/quality of the data obtained. However, recent advances in sensor-free detection of affect, based on deep learning methods, have substantially increased the quality of models [5], making interaction-based detectors a viable alternative. While these models are also not without limitations, their improved performance provides an alternative that facilitates near-continuous labeling at scale. As such, the recent advent of higher-quality detectors introduce the opportunity to study affect dynamics and affective chronometry with fine levels of granularity at scale.

In this paper, we present research studying affect dynamics and affective chronometry with the use of deep learning sensor-free affect detectors. We report the affect dynamics and chronometry for four commonly-studied affective states: engaged concentration [7] (also referred to as engagement, flow, and equilibrium), boredom [7][19], confusion [6][16], and frustration [16][23]. We investigate these relationships in the real-world learning of just under a thousand students, and compare our findings to prominent foundational research [9][10].

2. PREVIOUS WORK

The theoretical model of affective dynamics proposed by D'Mello and Graesser [10] has become widely recognized in the study of affective state transitions. The model proposes a set of theoretically hypothesized transitions that have emerged through the study of student affect, as illustrated by the simplified representation of the model in Figure 1. While the full model observes numerous affective states including surprise and delight, we restrict the analysis in this paper to the key affective states of engaged concentration, boredom, confusion, and frustration.

The model hypothesizes that specific transitions between affective states are particularly common. In this model, a student commonly begins in a state of equilibrium (i.e. flow or engaged concentration). The student remains in this state until novelty or difficulty emerges, at which point the student may transition to confusion. The student may transition back to engaged concentration by resolving this confusion, possibly experiencing delight upon the way. Alternatively, the student my transition from confusion to frustration, at which point the model suggests that the student is unlikely to transition back to the more productive cycle of engaged concentration and confusion; instead, the student is more likely to transition from frustration to boredom. As such, while students may be expected to oscillate between certain adjacent states in the model, the model suggests that it is unlikely for students to transition to unconnected states as depicted in Figure 1.

The model has been explored in several studies [27][8] observing differences in student affect, and has become influential to other research studying affect dynamics in the context of other constructs such as gaming the system [26]. Other studies prior to the publication of this model also studied affective dynamics [1][29]. While the specific affective states studied across these projects vary, the four affective states studied in this work are among the most commonly observed in this area of research. However, work in other paradigms also exists; for example, Redondo [25] attempted to identify when a student's affect shifts from increasingly positive to becoming more negative, or vice-versa, in self-report Likert scale data, finding that unexpectedly positive or negative affect typically indicated a shift in overall affective trajectory. However, she did not compare the prevalence of turning points found to overall base rates of affect, or analyze the chronometry of the sequences she studied. In general, across these papers, estimates of student affect have been collected through a range of methodologies including, most commonly, quantitative field observations (QFOs) [13][12][26][20], but also through self-reports in conjunction with post-hoc judgements of recorded video [3][4].

While there have been a large number of projects investigating affective dynamics, there has been substantially less research pertaining to affective chronometry. The study of affective chronometry is at times seen in affective dynamics papers. Among the papers investigating affective dynamics, several studies, including that of Baker, Rodrigo, and Xolocotzin [1] have found that state self-transitions, where the student is in the same affective state in one observation as in the previous observation, were often statistically significantly more likely than chance. This suggests that students in each state do tend to persist for at least the duration of the time interval between observations (1 minute in that article); however, this paper did not observe the chronometry beyond this interval. In foundational work in this area, D'Mello and Graesser [9] investigated the duration of different affective states, proposing a methodology with which to evaluate the "half-life," or decay of individual affective states experienced by students. Using a computer-based system known as AutoTutor, the authors used a combination of self-reports of the students and expert and peer judgments of student affect made using recorded video in order to measure and evaluate the length of time students commonly remained in each experienced affective state. However, that work was conducted on a relatively small number of subjects working on AutoTutor in a lab setting, on a task not related to their studies. It is therefore unclear whether the findings obtained in that context will generalize to data from a classroom environment where students are working on authentic educational tasks. The same methodology for measurement and evaluation of affective chronometry as presented in that work will be applied here to understand and compare affective chronometry – however, instead of using self-report, this project will utilize sensor-free detectors of affect applied to data collected from real students working in classroom environments.

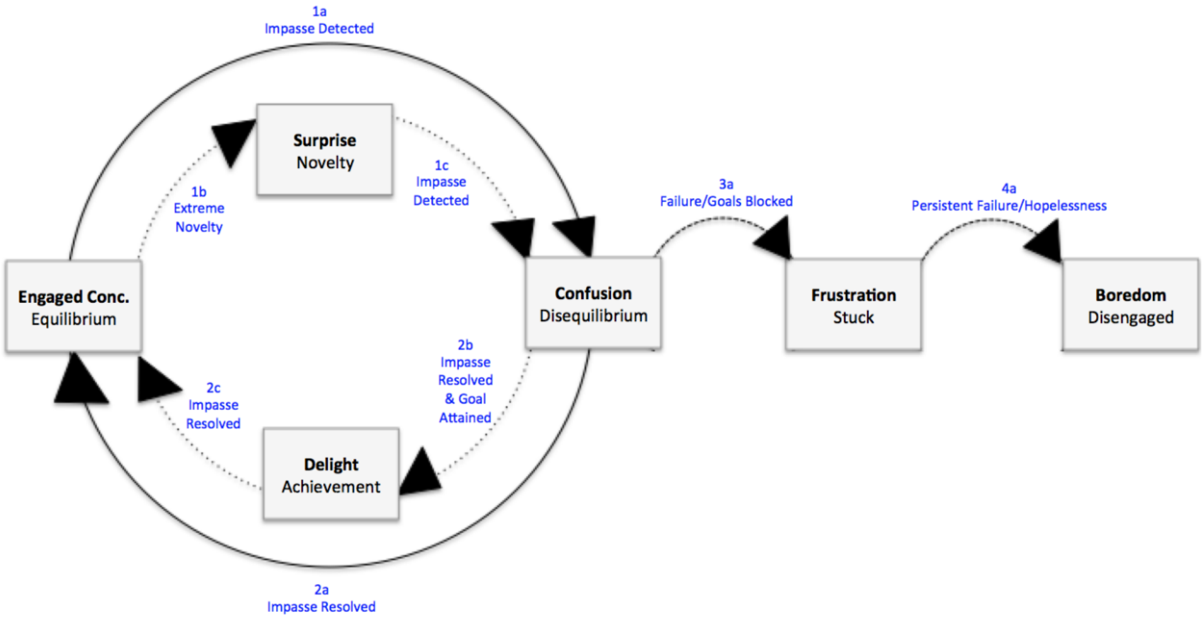


Figure 1: The proposed theoretical model of affect dynamics as presented by D'Mello and Graesser [10]

2.1 Detectors of Student Affect

We apply the sensor-free detectors of student affect previously described in Botelho et al. [5] to our data in order to study affective dynamics and chronometry. We use the same data set in this work from which the training set originally used in Botelho et al. [5] was sampled, to ensure maximum validity of the detectors. In applying the detectors to this data set, we determined that several minor adjustments needed to be made to the detectors, so that the training data set was aligned to the ground truth observations in a way that could be more easily applied to the unlabeled data. We also reduced the number of features used as input to the model building algorithm. The detectors were refit using this adjusted dataset and produced performance metrics comparable to the previous work (average AUC = .74, average Cohen's Kappa = 0.20).

As in Botelho et al. [5], these sensor-free detectors were developed using a long short term memory (LSTM) [15] network, a type of deep learning model designed for time series data. LSTM networks use a large number of learned parameters with internal memory that can model temporal trends within the data to make estimates that are better informed by previous time steps within the series. Although the initial training sample was imbalanced, the use of resampling did not improve model performance, and a min-max estimate scaling was used instead. The LSTM model is trained as a sequence-to-sequence model, meaning that it accepts an entire sequence of time steps as input and produces a sequence of outputs. These outputs are in the form of a sequence of estimates of the probability that each of four affective states of engaged concentration, boredom, confusion, and frustration are occurring at each 20-second time step, or "clip," within the data. We use this sequence of probabilities to study affective dynamics and chronometry – the details of these analyses are provided in later sections. The LSTM model was found to produce cross-validated AUC values that substantially outperformed prior sensor-free detectors, which had previously exhibited an average AUC = 0.66, developed using older algorithms with the same dataset [21][32]. In addition, LSTM models are designed to exploit the temporal character of the data, suggesting that they will be able to model temporal changes and transitions between affective state better than a model that treats each 20-second clip of student behavior as an independent sample.

3. METHODOLOGY

3.1 Dataset

The data[1] used in this work is comprised of action-level student data collected within the ASSISTments learning platform [14]. ASSISTments is a computer-based learning system used daily by thousands of students in real classrooms (over 50,000 a year) and hosts primarily middle school math content. The system has been used in several previous papers to study student affect, in many cases using sensor-free detectors of student affect.

Within this paper, we utilize a dataset originally used to develop sensor-free automated detectors of student affect. Detectors were originally developed using data collected by conducting field observations of student affect as 838 students used ASSISTments. 3,127 20-second field observations were collected in total, with gaps between one and several

[1] The data used in this work is made available at http://tiny.cc/EDM2018_affectdata

minutes between observations of the same student. For this paper, we analyze the entire data set of interaction for those 838 students on the days when observation occurred, 48,276 20-second segments of student behavior in total. We format the data in terms of 20-second segments of behavior in order to use the sensor-free detectors of affect, which were developed at this grain size (in line with the original field observations, which were conducted at the same grain size). The original training data set was highly imbalanced, with approximately 82% of observations coded as engaged concentration, 10% coded as boredom, 4% coded as confused, and 4% coded as frustration. This imbalance is consistent with previous research on the prevalence of these affective categories in systems such as ASSISTments.

The sensor-free LSTM detectors were applied to this dataset, providing an estimate of the probability of each of the four observed affective states for each of the 20-second segments of behavior within the system. The ground-truth labels used in model training are removed from this dataset and instead are replaced with the estimates produced by the sensor-free detectors. We replaced the ground-truth labels with the detector outputs so that the data would be comparable across all of the 48,276 observations.

3.2 Affect Dynamics

The estimates produced by the sensor-free detectors, when applied to the analysis dataset, are used to observe which transitions between affective states are frequent and statistically significantly more likely than chance. As is described in the previous section, the model produces four continuous-valued estimates corresponding with the 4 affective states of engaged concentration, boredom, confusion, and frustration. However, these estimates must be discretized and reduced to a single label describing the most likely affective state exhibited by the student at each time step. It is not sufficient to simply conclude that the most probable affective state (e.g. the affective state with the highest confidence) is the current affective state. For example, the model may predict very small values for all four affective states.

Instead, we first select a threshold that indicates that a specific affective state is likely occurring during a specific clip. We use a threshold of 0.5, defining a value above this threshold to be indicative of the presence of that corresponding affective state for the time step. 0.5 is a reasonable threshold as the detectors were previously run through a min-max scaling of the model outputs to remove majority class bias (cf. [5]). However, there exists the possibility, as expressed in the example above, that no estimate across the four affective states surpasses this defined threshold. In such cases, a fifth “Neutral/Other” affective state is introduced to represent that none of the affective states we are studying is occurring; this state has been included in similar previous analyses of affect dynamics as well ([13][12][29][27][4][9]). Conversely, it is possible for more than one estimate across the four outputs to surpass the defined threshold. In this unusual case (less than 1% of our data), no single affective state label can be applied and this clip (and transitions from and to this clip) is omitted from the subsequent analyses.

Once all estimates have been classified as either a single affective state or the neutral state, transitions between these states within each student are computed. As in [10], we omit self-transitions where the student remains in their current affective state; these are instead represented through affective chronometry (see next section). We report D’Mello’s L [11] as a measure of the commonality of each possible transition from a source affective state to a destination affective state along with a corresponding p-value denoting the probability of this frequency of transition being obtained by chance. The D’Mello’s L metric can be interpreted in a similar manner to Cohen’s kappa, describing the degree to which each transition is more (or less) likely than would be expected according to the overall proportion of occurrence of the destination affective state across all cases. Values of D’Mello’s L below zero are less likely than chance; values above zero represent the percent more likely than chance the finding is. In other words, a D’Mello’s L of 0.4 represents a transition that occurs 40% more often than would be expected from the destination state’s base rate. We compute statistical significance of these transitions using the method originally proposed in [11] – D’Mello’s L is computed for each student and transition, and then the set of transitions is compared to 0 using a one-sample two-tailed t-test. Benjamini and Hochberg’s [2] correction is used to control for the substantial number of statistical comparisons conducted.

3.3 Affective Chronometry

Our methodology for affective chronometry closely follows that of D’Mello and Graesser [9], with whom we compare our findings. In their analysis, the rate of decay was calculated as a probability of each state persisting over a 60-80 second window, using affect labels aggregated across multiple observation methods including the use of self-reports and both peer- and expert-observers. The probability that each affective state persisted (i.e. $\Pr(\mathrm{E}_t = \mathrm{E}_{t+20})$) was computed for 20 second intervals within that window.

The analysis in this paper uses the same discretized affect labels described in the previous section, transforming a sequence of sets of four probabilities to a single most-likely affective state per clip. The sequence of labels is broken into a set of episodes of each affective state, where an episode describes a series of non-transitioning affect that starts when the student transitions into the state and ends when the student transitions out of the state. A cumulative sum of time, in seconds, is calculated for each episode to measure how long each student remained in each affective state. With this value, a probability that a state will persist beyond a defined number of seconds can be calculated.

Due to the nature of our affect detection approach, persistence is estimated in 20 second intervals. At each interval, the probability that a student remains in eachtheir current affective state is calculated for durations up to 300 seconds, or 5 minutes. The resulting 16 probabilities (for durations of 0, 20, 40, ... , 300 seconds) can then be used to compare the rates of decay across each of the observed affective states.

4. RESULTS

4.1 Observing Affect Dynamics

The affective state transitions, measured by D’Mello’s L, are reported in Table 1 with accompanying significance. Aside from those transitions that occur to/from the neutral/other

Table 1: The transitions between affective states. D'Mello's L values are shown. Transitions that are statistically significantly more likely than chance, after Benjamini and Hochberg's post-hoc correction, are denoted *.

From State	To State	D'Mello's L	p-value
Engaged Concentration	Engaged Concentration	—	—
	Boredom	0.260*	<0.001
	Confusion	0.004	0.136
	Frustration	-0.12*	0.012
	Neutral/Other	0.481*	<0.001
Boredom	Engaged Concentration	0.194*	<0.001
	Boredom	—	—
	Confusion	-0.004	0.208
	Frustration	0.036*	<0.001
	Neutral/Other	0.235*	<0.001
Confusion	Engaged Concentration	0.341*	0.006
	Boredom	-0.127*	<0.001
	Confusion	—	—
	Frustration	-0.026*	0.001
	Neutral/Other	-0.156	0.157
Frustration	Engaged Concentration	0.279*	<0.001
	Boredom	-0.107*	<0.001
	Confusion	0.008	0.391
	Frustration	—	—
	Neutral/Other	0.279*	<0.001
Neutral/Other	Engaged Concentration	0.753*	<0.001
	Boredom	-0.057*	<0.001
	Confusion	0.003	0.302
	Frustration	0.015*	0.007
	Neutral/Other	—	—

state, the most common significant transition appears to occur between confusion and engaged concentration, followed by that of frustration to engaged concentration. Contrary to the theoretical model proposed by D'Mello and Graesser [10], significant transitions are found between engaged concentration and boredom as well as from boredom to engaged concentration. The findings suggest that students do not transition between these states through others as in the proposed theoretical model, but can occur directly.

It is further illustrated in the table that no state is found to transition to confusion more likely than chance, for which there are several possible explanations. Confusion was the least-frequently detected state as estimated by the sensor-free model (under 1.0% of the dataset). As such, it is likely that there simply were not enough instances of detected confusion in the data to produce significant results, possibly because the model had difficulty detecting confusion, contributing to an under-sampling of this state as estimated by the model.

These positive and significant transitions as identified by Table 1 are illustrated in Figure 2 for better comparison to the theoretical model depicted in Figure 1. Not only do the already-identified transitions become clearer, the number of transitions occurring to and from the neutral/other state, listed simply as "no label" in that figure, are also made prominent. As described in the generation of this fifth state, this represents those estimates where no model estimates across the four affective states exceeded the defined threshold. It is important to note that this state may not be a single state at all, but rather comprehensively represents all other affective states exhibited by students that are not observed in the analysis. As such, it is difficult to make meaningful claims or draw significant conclusions regarding transitions occurring to or from this state.

The divergence of the emerging transitions and the theoretical model indicate that there are fewer oscillations that are detected by the machine-learned method. While not included in the theoretical model, D'Mello and Graesser propose in the same work [10] that oscillations can occur between all adjacent affective states within the graph under certain conditions, but that is certainly not the case as seen in Figure 2 gained from the empirical results of this work. This suggests that the learned model finds that students do

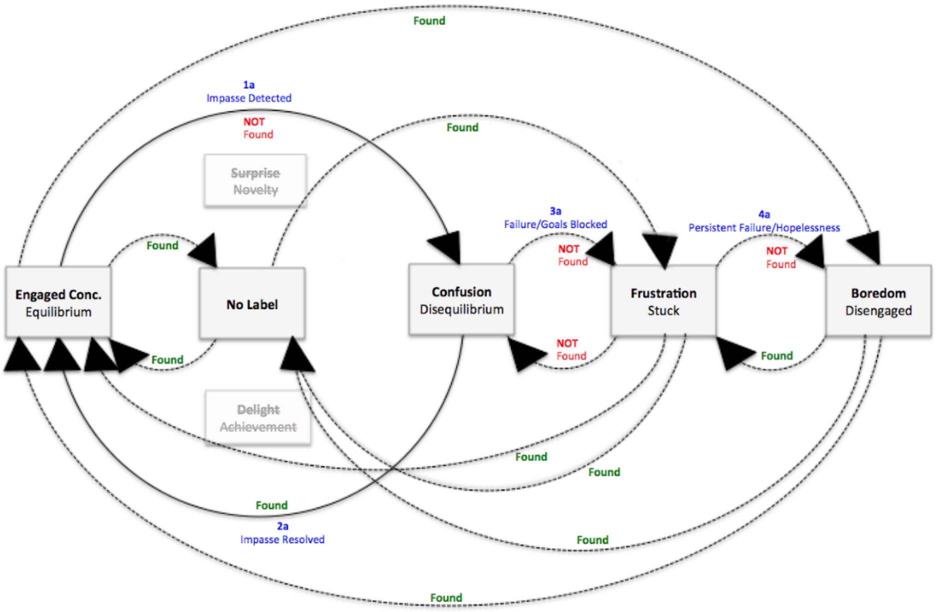


Figure 2: The resulting positive and significant affect transitions as compared to the D'Mello and Graesser [10] theoretical model.

not commonly transition back and forth between states such as confusion and frustration as often as hypothesized by the theoretical model, but no other such cases emerge.

4.2 Observing Affective Chronometry

The results of our affective chronometry analysis illustrate the length of time students commonly spend in each affective state before transitioning to either another observed state or the neutral/other state. The results of this analysis, depicted in Figure 3, show notable differences in affective half-life between affective states. Engaged concentration and boredom exhibit much more gradual declines as opposed to both confusion and frustration which both exhibit steep and rapid decay. Just as was done in the previous work of D'Mello and Graesser [9], the decay can be quantified by fitting an exponential function to each of the observed states. Again, as the neutral/other state may comprehensively represent multiple states that are not measured in this work, this state is not included in the analyses of affective chronometry; if included, the results may simply illustrate an average decay over non-included affective states.

The value of decay for each state, as calculated by fitting an exponential curve to each states probability of persisting (Pr(No Change)) over time. Engaged concentration (decay = -0.003) and boredom (decay = -0.004) are found to have similarly gradual decay as compared to that of the remaining two states. Frustration (decay = -0.01) and confusion (decay = -0.024) are found to decay significantly faster. Of the studied states, only confusion is found to fail to persist past 5 minutes.

While the affective decay of engaged concentration, boredom, and frustration follow the general trend found by the work of D'Mello and Graesser in previous work [9], confusion deviates from this alignment. This difference is illustrated by Figures 4 and 5. Figure 4 illustrates the plotted exponential fit lines that were learned from the estimates produced by the sensor-free detectors. For comparison, Figure 5 illustrates the plotted exponential decay, as reported in Table 1 of D'Mello and Graesser [9]. From this, it becomes apparent that confusion is found to exhibit similar decay patterns to that of engaged concentration and boredom, being more gradual over time, than that of frustration.

The other distinctive difference that emerges from the comparison of Figures 4 and 5 is that of the average time for decay across all affective states. This suggests that the average time that students remain in any affective state, as determined by the sensor-free model, is consistently longer than those found in D'Mello and Graesser [9]. The previous work reports that students rarely remained in a single state for longer than 60 seconds, and, following the learned exponential curve in Figure 5, no state seems to persist beyond 3 minutes, with most states reaching a probability of persisting close to 0 long before that time point. In comparison, each of the affective states, with the exception of confusion, are found to persist past the 5 minute time point, with engaged concentration and boredom seemingly persisting significantly beyond this point. Even in considering the 60 second timeframe, the fastest decaying state of confusion exhibits students persisting beyond this interval.

The divergence of the decay rates as exhibited by the estimates of the sensor-free model and those of the empirical findings reported in [9] may be due to a combination of differences between the two works. One possible explanation is the difference in learning contexts and the different learning interactions being studied in each of the two works. In this work, for example, the students comprising

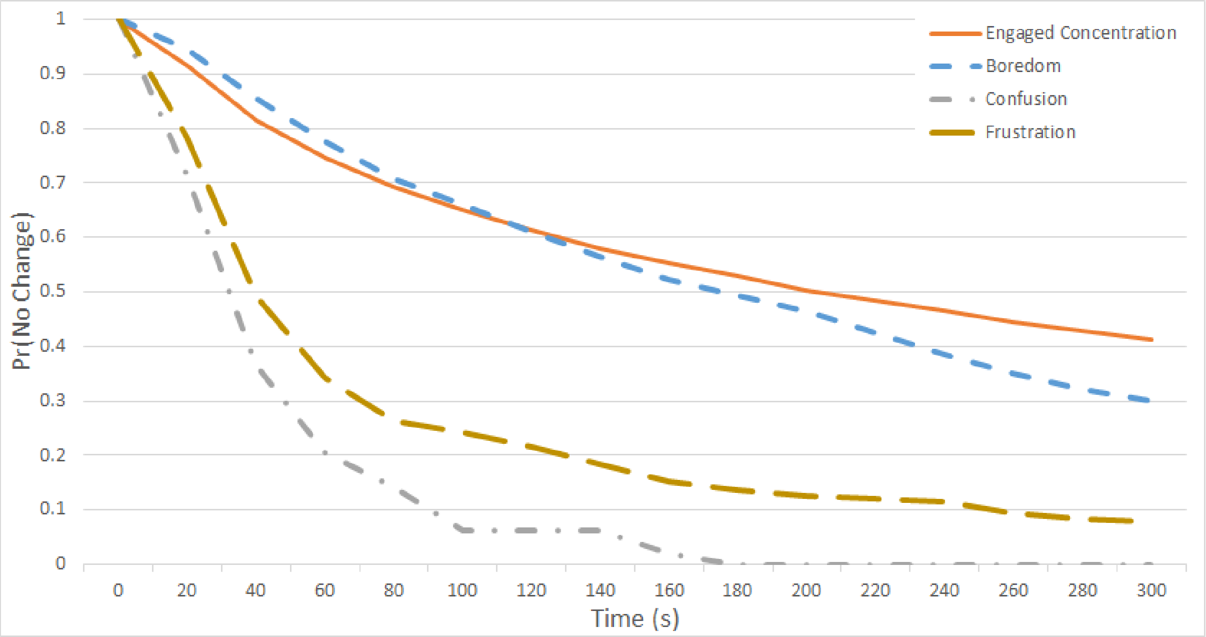


Figure 3: The probability of a student persisting in each affective state over time.

the dataset were in a classroom environment interacting with the computer-based system of ASSISTments. The previous study reported by [9], had students interacting with different software, namely that of AutoTutor, and also took place in a controlled lab setting. The domain of study also exhibits differences in that the students in AutoTutor were answering questions pertaining to computer literacy that are described as requiring students to answer in several sentences. The students using ASSISTments, however, were middle school students working on math content. The differences between both the content and the environment could have a distinct effect on the states of affect exhibited by students as well as the length of time students persist in each affective state.

5. DISCUSSION AND FUTURE WORK

The current work presents, to the knowledge of the authors, the first application of sensor-free affect detectors to study affect dynamics and affective chronometry. In studying affective dynamics, we can compare our results to a past theoretical model of affect dynamics proposed by D'Mello and Graesser [10], as well as other past empirical work. In affective chronometry, we can compare our results to past work [9], also by D'Mello and Graesser. The resulting model of affect dynamics produced by the application of sensor-free detectors shares little with the theorized model in regard to the significant transitions that emerged. Most notably, our model suggests oscillations between engaged concentration and boredom which are hypothesized not to occur significantly in the theorized model; it has been found in other empirical work, however, that transitions between engaged concentration and boredom do appear [3][4]. The model of affective chronometry finds a similar pattern to D'Mello and Graesser in terms of which affective states are shorter and longer, but we find that all affective states last longer in our data set than in their previous work.

The application of sensor-free detectors to the study of student affect provides the opportunity to study how such affect is exhibited in students at greater scale and at second-by-second levels of granularity. In addition, automated detectors are a less intrusive method of data collection than more traditional methods. As the detectors utilize only data recorded from computer-based systems, they can estimate a student's affective state without interrupting their work, as can be the case with self-reporting methods, and does not hold a risk of observer effects where students change their behavior due to the presence of a human coder. The method also does not require the use of additional technology such as physical and physiological sensors that may be difficult to deploy in classrooms at scale. Given the greater scale facilitated by automated affect detectors, future research may be able to study not just overall affective dynamics and chronometry but how dynamics and chronometry vary between different activities, different student populations, and even at different times of day. The better understanding of affective dynamics and chronometry that this may afford may have several benefits. Understanding a system's affective dynamics may be useful for encouraging positive transitions and suppressing negative transitions. Understanding affective chronometry may help us understand when negative emotion is problematic. Although some confusion is associated with positive learning outcomes [17], extended confusion is associated with worse student performance [18]. Understanding whether a student's confusion or frustration lasts longer than the expected duration may indicate that a student is struggling and is in need of intervention.

As the scale of the application of automated detectors increases for the study of affective dynamics, the means of evaluating common transitions will likely need to evolve as well. After a certain data set size, all transitions will become significant. Even in this paper, with a relatively limited data

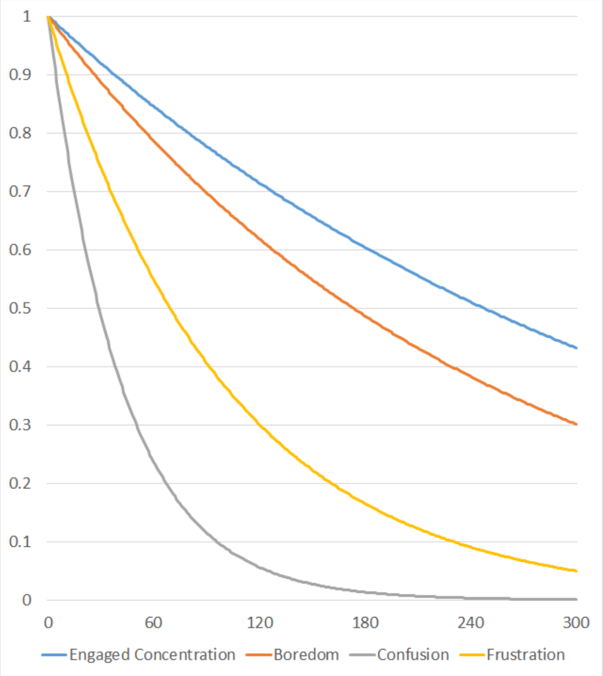


Figure 4: The plotted exponential decay of each affective state as estimated by the sensor-free affect detectors.

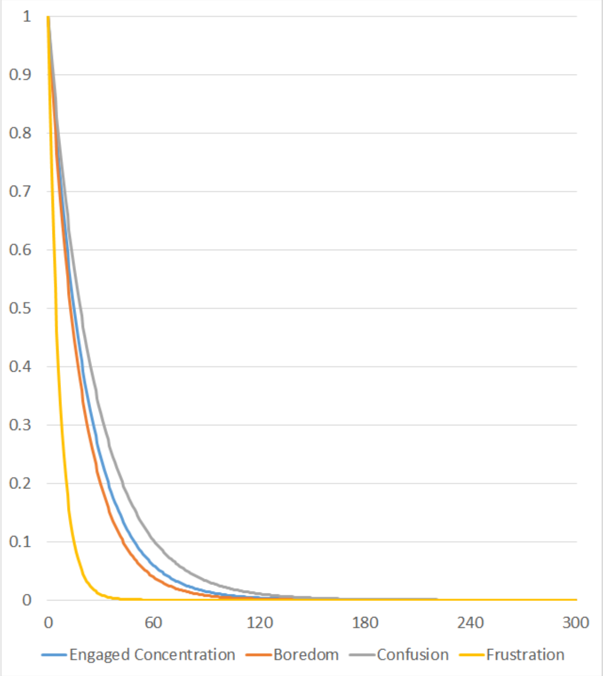


Figure 5: The plotted exponential decay of each affective state as reported in Table 1 of D'Mello and Graesser [9]

set, fairly low values of D'Mello's L reached statistical significance. Future work may need to explore new methods of identifying and evaluating affect dynamics, perhaps by simply exploring reasonable means of leveraging D'Mello's L as a measure of magnitude to identify meaningfully frequent links, not just those that are simply statistically significantly more likely than chance.

There are potential limitations to the current work that may be addressed by future research in this area. First, while the sensor-free detectors used in this work, as presented in [5], exhibit significantly superior performance to previous developed detectors with regard to AUC, improving the performance of these models further may help to improve transition and chronometry estimates, particularly of the less common labels of confusion and frustration. Utilizing methods to supplement less-frequently occurring labels of student affect (though the common method of resampling did not, in fact, enhance these detectors) or utilizing unlabeled data to better inform model estimates through co-training may improve model performance and produce more accurate measurements of affect dynamics and affective chronometry. It also may make sense to use different confidence thresholds for different affective states to adjust for the differences in the conservatism of different detectors that emerge from having different base rates.

Although consisting of a small portion of the data used in this work, the analyses did not include cases of co-occurring labels as estimated by the model. The estimates produced by the sensor-free detectors, even when the ground truth labels used to train such detectors did not observe co-occuring affective states themselves, is able to produce such cases, providing the opportunity to observe such cases in future work. Identifying which states are likely to co-occur, as well as include such cases in analyses of state transitions and affect state decay, will help to gain a better understanding of the relationships between affective states as well as to student performance.

A final opportunity for future work is in regard to observing affect dynamics and chronometry in experimental settings, as in the case of randomized controlled trials (RCTs). Several works have used analyses of state transitions to observe differences in affect exhibited between experimental conditions [27][8]. As the training set used to develop affect detectors does not contain experiment data, it is at this time uncertain if they generalize to behaviors exhibited outside of normal usage of the learning platform. Future work can observe how well such detectors generalize to such populations of users and samples.

6. ACKNOWLEDGMENTS

We thank multiple current NSF grants (IIS-1636782, ACI-1440753, DRL-1252297, DRL-1109483, DRL-1316736, DGE-1535428 & DRL-1031398) , the US Dept. of Ed (IES R305A120125 & R305C100024 and GAANN), and the ONR.

7. REFERENCES

[1] R. S. Baker, M. M. T. Rodrigo, and U. E. Xolocotzin. The dynamics of affective transitions in simulation problem-solving environments. In *International Conference on Affective Computing and Intelligent Interaction*, pages 666–677. Springer, 2007.

[2] Y. Benjamini and Y. Hochberg. Controlling the false

discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, pages 289–300, 1995.

[3] N. Bosch and S. D'Mello. Sequential patterns of affective states of novice programmers. In *The First Workshop on AI-supported Education for Computer Science (AIEDCS 2013)*, pages 1–10, 2013.

[4] N. Bosch and S. D'Mello. The affective experience of novice computer programmers. *International Journal of Artificial Intelligence in Education*, 27(1):181–206, 2017.

[5] A. F. Botelho, R. S. Baker, and N. T. Heffernan. Improving sensor-free affect detection using deep learning. In *International Conference on Artificial Intelligence in Education*, pages 40–51. Springer, 2017.

[6] S. Craig, A. Graesser, J. Sullins, and B. Gholson. Affect and learning: an exploratory look into the role of affect in learning with autotutor. *Journal of educational media*, 29(3):241–250, 2004.

[7] M. Csikszentmihalyi. Flow. the psychology of optimal experience. new york (harperperennial) 1990. 1990.

[8] S. D'Mello and A. Graesser. Modeling cognitive-affective dynamics with hidden markov models. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 32, 2010.

[9] S. D'Mello and A. Graesser. The half-life of cognitive-affective states during complex learning. *Cognition & Emotion*, 25(7):1299–1308, 2011.

[10] S. D'Mello and A. Graesser. Dynamics of affective states during complex learning. *Learning and Instruction*, 22(2):145–157, 2012.

[11] S. D'Mello, R. Taylor, and A. Graesser. Monitoring affective trajectories during complex learning. In *29th Annual Meeting of the Cognitive Science Society*, pages 203–208. Springer, 2012.

[12] T. F. G. Guia, M. M. T. Rodrigo, M. Dagami, C. Marie, J. O. Sugay, F. J. P. Macam, and A. Mitrovic. An exploratory study of factors indicative of affective states of students using sql-tutor. *Research & Practice in Technology Enhanced Learning*, 8(3), 2013.

[13] T. F. G. Guia, J. O. Sugay, M. M. T. Rodrigo, F. J. P. Macam, M. M. C. Dagami, and A. Mitrovic. Transitions of affective states in an intelligent tutoring system. *Proceedings of the Philippine Computing Society*, pages 31–35, 2011.

[14] N. T. Heffernan and C. L. 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.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[16] 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, 2001. Proceedings. IEEE International Conference on*, pages 43–46. IEEE, 2001.

[17] B. Lehman, S. D'Mello, and A. Graesser. Confusion and complex learning during interactions with computer learning environments. *The Internet and Higher Education*, 15(3):184–194, 2012.

[18] Z. Liu, V. Pataranutaporn, J. Ocumpaugh, and R. Baker. Sequences of frustration and confusion, and learning. In *Educational Data Mining 2013*. Citeseer, 2013.

[19] M. Miserandino. Children who do well in school: Individual differences in perceived competence and autonomy in above-average children. *Journal of educational psychology*, 88(2):203, 1996.

[20] J. Ocumpaugh, J. M. Andres, R. Baker, J. DeFalco, L. Paquette, J. Rowe, B. Mott, J. Lester, V. Georgoulas, K. Brawner, et al. Affect dynamics in military trainees using vmedic: From engaged concentration to boredom to confusion. In *International Conference on Artificial Intelligence in Education*, pages 238–249. Springer, 2017.

[21] J. Ocumpaugh, R. Baker, S. Gowda, 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.

[22] Z. A. Pardos, R. S. Baker, M. San Pedro, S. M. Gowda, and S. M. Gowda. 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, 2014.

[23] B. C. Patrick, E. A. Skinner, and J. P. Connell. What motivates children's behavior and emotion? joint effects of perceived control and autonomy in the academic domain. *Journal of Personality and social Psychology*, 65(4):781, 1993.

[24] M. O. Pedro, R. Baker, A. Bowers, and N. Heffernan. Predicting college enrollment from student interaction with an intelligent tutoring system in middle school. In *Educational Data Mining 2013*, 2013.

[25] G. N. Redondo. Turning points en las trayectorias emocionales de estudiantes en un contexto desafiante de aprendizaje experiencial: una aproximación dinámica. In *Quintas Jornadas de Jóvenes Investigadores de la Universidad de Alcalá: Humanidades y Ciencias Sociales*, pages 245–254. Servicio de Publicaciones, 2016.

[26] M. M. T. Rodrigo, R. d. Baker, J. Agapito, J. Nabos, M. Repalam, S. Reyes Jr, and M. San Pedro. The effects of an embodied conversational agent on student affective dynamics while using an intelligent tutoring system. *IEEE Transactions on Affective Computing*, 2(4):18–37, 2011.

[27] M. M. T. Rodrigo, R. S. Baker, J. Agapito, J. Nabos, M. C. Repalam, S. S. Reyes, and M. O. C. San Pedro. The effects of an interactive software agent on student affective dynamics while using; an intelligent tutoring system. *IEEE Transactions on Affective Computing*, 3(2):224–236, 2012.

[28] M. M. T. Rodrigo, R. S. Baker, M. C. Jadud, A. C. M. Amarra, T. Dy, M. B. V. Espejo-Lahoz, S. A. L. Lim, S. A. Pascua, J. O. Sugay, and E. S. Tabanao. Affective and behavioral predictors of novice programmer achievement. In *ACM SIGCSE Bulletin*, volume 41, pages 156–160. ACM, 2009.

[29] M. M. T. Rodrigo, G. Rebolledo-Mendez, R. Baker,

B. du Boulay, J. Sugay, S. Lim, M. Espejo-Lahoz, and R. Luckin. The effects of motivational modeling on affect in an intelligent tutoring system. In *Proceedings of International Conference on Computers in Education*, volume 57, page 64, 2008.

[30] M. O. San Pedro, J. Ocumpaugh, R. S. Baker, and N. T. Heffernan. Predicting stem and non-stem college major enrollment from middle school interaction with mathematics educational software. In *EDM*, pages 276–279, 2014.

[31] R. L. Solomon and J. D. Corbit. An opponent-process theory of motivation: I. temporal dynamics of affect. *Psychological review*, 81(2):119, 1974.

[32] Y. Wang, N. T. Heffernan, and C. Heffernan. Towards better affect detectors: effect of missing skills, class features and common wrong answers. In *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge*, pages 31–35. ACM, 2015.

Predicting Quitting in Students Playing a Learning Game

Shamya Karumbaiah
University of Pennsylvania
3700 Walnut St
Philadelphia, PA 19104
+1 877 736-6473
shamya@upenn.edu

Ryan S. Baker
University of Pennsylvania
3700 Walnut St
Philadelphia, PA 19104
+1 215 573-2990
ryanshaunbaker@gmail.com

Valerie Shute
Florida State University
1114 West Call Street
Tallahassee, FL 32306-4453
+1 850 644-8785
vshute@fsu.edu

ABSTRACT

Identifying struggling students in real-time provides a virtual learning environment with an opportunity to intervene meaningfully with supports aimed at improving student learning and engagement. In this paper, we present a detailed analysis of quit prediction modeling in students playing a learning game called Physics Playground. From the interaction log data of the game, we engineered a comprehensive set of aggregated features of varying levels of granularity and trained individualized level-specific models and a single level-agnostic model. Contrary to our initial expectation, our results suggest that a level-agnostic model achieves superior predictive performance. We enhanced this model further with level-related and student-related features, leading to a moderate increase in AUC. Visualizing this model, we observe that it is based on high-level intuitive features that are generalizable across levels. This model can now be used in future work to automatically trigger cognitive and affective supports to motivate students to pursue a game level until completion.

Keywords

disengagement, learning games, quit prediction, adaptive intervention, personalized learning, physics education

1. INTRODUCTION

In the past couple of decades, education researchers and developers have looked into using digital games as vehicles for learning in a range of domains [19]. Learning games are designed with the goal of keeping students engaged in a fun experience while also focusing on their learning. Well-designed games help build intrinsic motivation in players, which they sustain throughout the process by keeping the player in a state of deep engagement or flow [7].

For a successful learning experience, Gee [14] emphasizes that the game must focus on the outer limits of the student's abilities, making it hard yet doable – Csikszentmihalyi similarly suggests that optimal flow is achieved when student ability is matched with game difficulty [7]. Although some researchers have argued that the difficulty associated with the highest engagement is different than the difficulty associated with the highest learning [20], the goal of good game design must be to promote both engagement and learning.

The challenge, then, must be to maintain the high difficulty associated with learning without compromising engagement to a degree that the student becomes highly frustrated or worse, gives up [e.g. 20]. After all, if a student gives up, they typically do not continue learning from the game (at least, not in the absence of reflective or teacher-driven discussion of the game – e.g. [28, 22]). Some students may quit a game level (or the entire game) only after protracted struggle. Others may quit the level immediately and search for an easier level, a behavior tagged as the "soft underbelly strategy" [1]. Both responses to difficulty should be addressed in an optimal learning game.

To prevent students from giving up, most serious games in education include immediate feedback and interventions aimed to improve the learner experience [26]. When the student is struggling, a relevant and timely intervention could keep the student motivated and prevent frustration from leading the student to give up. A struggling student may also benefit from an intervention that prevents them from wheel-spinning [4], playing for substantial amounts of time without making progress.

However, even though scaffolding may be beneficial to a struggling student, it may be undesirable – even demotivating and harmful to learning – if the student is provided with scaffolding when he or she does not need it [9]. As such, it may be valuable to detect struggling during games that can benefit from an intervention. In that fashion, scaffolding can be provided to students who need it but withheld where it is unnecessary and may be counterproductive. The goal of this paper, then, is to detect whether a student is likely to give up and quit a level in progress. We do so in the context of Physics Playground [27], a game where students learn physics concepts through interactive gameplay.

1.1 Related Work

There has been considerable interest in developing automated detectors of disengagement over the last decade. This work includes detectors for off-task conversation [2], mind wandering while reading [8], and gaming the system - where the student exploits the system to complete the task [3]. In the specific case of games, researchers have developed detectors for a variety of disengagement-related constructs, including whether the learner is engaging in behaviors unrelated to the game's learning goals [23], whether the student is genuinely trying to succeed in the game [10], and whether the learner is gaming the system [29]. One inherent challenge to much of the work to detect disengagement is the dependence on subjective human judgement for ground truth labels such as field observations, self-reports, and retrospective judgement. This makes it challenging to validate the model beyond the context of data collection. By contrast, predicting whether a student will quit has the advantage of only needing an objective ground truth label. This aspect of quit prediction makes it relatively less labor-intensive to validate a model in newer settings and diverse student population.

There has been past work to predict whether a student will quit within other types of online learning environments. In a lab experiment with a simple reading interface, an interaction-based detector was developed to predict if a student would quit an upcoming text based on the reading behavior of the student in the past text [21]. There has also been considerable attention to the issue of quit prediction (sometimes referred to as dropout or stop-out) in the context of massive open online courses (MOOC), due to the high attrition rate in MOOCs. In one of the studies [31], researchers conducted social network analysis (based on discussion forum participation) and survival analysis to predict student dropout from an ongoing Coursera class. Another study [15] detected at-risk students based on their engagement with video lectures and assignments and their performance in the assignments. One important aspect to some of this MOOC work is that the detectors have been used to drive interventions. For instance, an automatic survey intervention was built based on a MOOC dropout classifier by researchers at HarvardX [30]. They observed that the surveys appeared to increase the proportion of students thought to have dropped out who chose to return to the course.

1.2 Context/Setting

Physics Playground[1] (PP; formerly known as Newton's Playground) [27] is a two-dimensional game, developed to help secondary school students understand qualitative physics related to Newton’s laws of force and motion, mass, gravity, potential and kinetic energy, and conservation of momentum. The player draws *objects* on the screen, often simple machines or *agents* to guide a green ball to hit a red balloon (goal) by using a mouse and drawing directly on the screen.

The agents in the game were as follows: A ramp is any line drawn that helps to guide the ball in motion (e.g., such as a line that prevents the ball from falling into a hole). A lever rotates around a fixed point (pins are used to fix an object on- screen), and is useful for moving the ball vertically. A swinging pendulum directs an impulse tangent to its direction of motion, and is usually used to exert horizontal force. A springboard stores elastic potential energy provided by a falling weight, and is useful for moving the ball vertically. Such weights are called freeform objects whose mass is determined by the density of the drawn object.

Any solution that solves the problem receives a silver badge; a solution that solves the problem with a minimal number of objects receives a gold badge. Problems are designed so that receiving a gold badge typically requires a specific application of an agent or simple machine. Laws of physics apply to the objects drawn by the player. There are seventy-four levels in total across seven playgrounds. Each level contains fixed and movable objects. The player analyzes the givens (what he/she sees on the screen) and sketches a solution by drawing new objects on the screen (see Figure 1). All objects in the game obey the basic rules of physics relating to gravity and Newton’s laws, and each level is designed to be optimally solved by particular agents. PP is nonlinear; students have complete choice in selecting playgrounds and levels.

The goal of quit prediction is to identify potential learning moments for a struggling student in the game where a cognitive support could support the student in developing their emerging understanding of key concepts and principles.

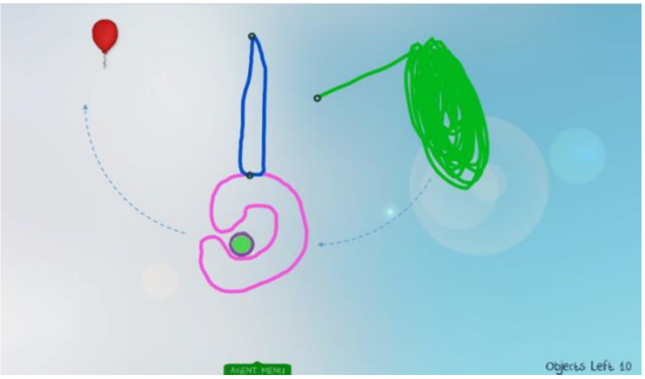


Figure 1. An example level in physics playground being solved with a pendulum agent (drawn in green by the student). The dashed blue (marked for illustration; not shown in the game) line traces the trajectory of the pendulum when released and that of the ball to the balloon after the pendulum strikes.

2. METHODS

2.1 Data Collection

Participants consisted of 137 students (57 male, 80 female) in the 8th and 9th grades enrolled in a public school with a diverse population in a medium-sized city in the southeastern U.S. The game content was aligned with state standards relating to Newtonian Physics. The study was conducted in a computer-enabled classroom with 30 desktop computers over four consecutive days. On the first day, an online physics pretest was conducted, followed by two consecutive days of gameplay and a posttest on the fourth day. The pre-test and the post-test measured students’ proficiency in Newtonian physics. The software logged all the student interactions in a log file. In this paper, we focus on the data collected during the second and third days (where students were playing Physics Playground for 55 minutes each day).

Physics Playground log data capture comprehensive information on student actions and game screen changes as a time series with millisecond precision. One of the important fields in the log data is the *event*. It is used to construct most of the features used in our model. The value of this field categorizes the game moments into – a) game-related events like game start, and end; b) level-related events like start, pause, restart, and end; c) agent creation events like drawing of ramp, pendulum, level, springboard; d) play-related events like object drop, object erase, collision and nudge; e) between-level navigation events like menu-focus. We focus on level-related events, agent creation events, and play-related events for predicting whether a student will quit a specific level.

Some levels in PP can be solved by multiple agents (ramp, lever, pendulum, and springboard). For each of the relevant agents, students can get a silver or a gold badge based on how efficient their solution is. Hence, a student could be playing a level for the first time, replaying using a different agent, or replaying to get a better badge. We consider each of these visits to a level as separate instances of gameplay on that level and predict whether a student will quit the level during the student’s current visit. Each time a student exits a level, the log data marks the end of the visit with a level end event. This event can occur either when the student solves the level successfully (earns a badge; quit=0) or when the student exits a level without solving it (doesn’t earn a badge; quit=1).

[1] Link to play PP - https://pluto.coe.fsu.edu/ppteam/pp-links/

Within each visit, a student can restart a level multiple times without quitting the level. Restarting a level erases all the student-created objects and resets the ball and the other level-given objects back to their default positions. The ball also resets back to its original position each time it drops out of the screen. We identify this as a ball reset event.

2.2 Data Preparation

2.2.1 Data Pre-processing

Among the total of seventy-four levels in this version of the game, only thirty-four levels had data for at least fifty students. These levels were used for modelling (Table 1); the other levels did not have enough data to build level specific models (explained in section 2.3.1). Also, these higher levels are only reached by the most successful students, making this data of less interest for our research goal. After data pre-processing, we have 390,148 relevant events across all the students playing the chosen levels.

2.2.2 Feature Engineering

Feature engineering is an important step in the modeling pipeline that converts raw log data to a set of meaningful features. Many argue that the success of data mining approaches relies on thoughtful feature engineering [24]. For each data sample, we have engineered a total of 101 features of the four types listed below. In designing features, we endeavor to avoid using data about the student's future to interpret their behavior, since our goal is to predict their future outcome. Hence, all the features at any time step solely include the information from the past and the present.

a) *Student+Level+Visit* related features define a student's progress in their current visit to a level. They are recalculated at each event within the logs, and each row represents a single event or student action. There are multiple kinds of student+level+visit features: 1) A set of binary features denote the occurrence of an event (e.g., level restart, ball reset, and, the creation of an object). For these features, each row in the data represents a single event, so only one binary feature will have a value of 1 in any row; 2) A set of numerical features represent the current counts of all the actions taken by the student since the beginning of the visit. These include counts of objects and agents drawn and other relevant events (e.g., the number of springboards, freeform objects, pins, and ball nudges); 3) A set of features track higher-level game activities since the start of the visit (e.g., the number of level restarts and ball resets); 4) A set of temporal features (e.g., the time elapsed in the visit so far and the time elapsed since the last restart); and 5) A set of features that maintain the counts of currently active objects on screen since the drawn objects could drop off the screen or be erased by the student. There are a total of 27 student+level+visit related features. All of these features are updated after each relevant event (see section 2.1). In most cases, only a small subset of feature values change between consecutive data samples.

b) *Student+Level* related features define the student's experience with the level so far, across all the previous visits (recall that a student can replay a previously solved or unsolved level; see section 2.1). This includes high-level features like the number of visits to the level, the number of badges received in the past visits, the number of visits quit without solving, the overall number of pauses, and the total pause duration in the level overall. This also includes cumulative features that indicate past solution approaches (e.g., the total number of pendulums drawn in the past visits). There is a total of 17 such features. These are set to 0 for the first visit and is updated at the end of each consecutive visit to the level by the student.

c) *Student* related features define the student's progress through the game across all the levels played so far. These include counts like the total number of levels played, the number of levels quit, the number of levels involving a particular physics concept played so far (e.g., Newton's first law of motion, energy can transfer, properties of torque), and the number of levels solved using a particular agent. These also include an overall summary of gameplay attributes across the levels played so far (e.g., means and standard deviations of the number of visits, pause duration, time spent, and number of objects used across all the levels played so far). There are a total of 40 such features. The feature values start at zero for a new student and continue to get updated as the student proceeds playing more levels in the game.

d) *Level* related features define the inherent qualities of a particular level. There are two kinds of level-related features – 1) A set of ten features computed by taking averages and standard deviations of student-level features from all students who played that level (e.g., means and standard deviations of number of objects used, time taken, number of level restarts, and badges received in this level); and 2) A set of seven level-related features that do not require past student data. These include binary features for primary physics concept and agent(s) used for solving. There are a total of 17 level-related features. These features are pre-set at the game start and their values remain the same for all the students and all the visits to a particular level.

Upon exploring the relationship between the level-pause and level-end events, we noticed that in order to access the quit button, students need to pause the gameplay. Since level-pause is directly indicative of the outcome variable (though not all pauses lead to quitting), we have discarded any feature that is related to the occurrence of a pause event from the student+level+visit set of features and retained the pause-related features in the student+level set of features.

2.2.3 Aggregations

As we are predicting an outcome (quitting a level) that comes as the culmination of many actions, and that is likely to be predicted by patterns of inter-related actions rather than single actions (such as drawing a single object), we aggregate the data into 60-second clips [24], [5]. Since only student+level+visit (see Section 2.2.2 a) and student+level (see Section 2.2.2 b) features change with each event, these are the only features to be aggregated at the 60 second interval. The binary student+level+visit features are converted to integer features that count the occurrence of these events over the 60 second interval. For cumulative features like the total number of level restarts in the visit so far, the last value at the end of the 60 second window is retained. Similarly, for features indicating the current object counts and on-screen elements like current number of lever objects, the values of the last data sample in the 60 second interval are retained. The same approach is followed for the features corresponding to elapsed time, like time elapsed since level restart in the visit. After feature aggregation, we have a final sample size of 14,116 data points and a feature space of 101 dimensions.[2]

[2] The aggregated data (section 2.2.3) is made available at https://upenn.box.com/s/4ocucflaehd7c51lbxx96heikcjtcwz1

2.3 Model Training

The next step of the modeling process is to define the quit value for each data sample. We predict a binary label that represents whether the student quit a visit (without solving) or not. This variable can be operationalized in several fashions. One possible way to define the label value would be to only mark the last data sample of a visit before the student quits as representing quitting, as in the research on MOOC stop-out [30]. However, our goal is to be able to detect that a student is likely to quit early enough to prevent this behavior. Therefore, we label every 60-second data clip during a visit that is eventually quit as "quit". The overall class distribution in the data is 28.77% quit and 71.23% not-quit.

2.3.1 Level-specific Models Versus Level-agnostic Model

Within this paper, we consider two possible types of models for detecting quit: a) *level-specific models* which are trained on the data from a single level; and b) a *level-agnostic model* which is a single model trained on the data from all levels. One can see pros and cons to both approaches. We could expect level-specific models to be more accurate as the data is tailored to a narrower prediction context. However, using level-specific models necessitates having enough training data for all the levels. It also implies that detection will be unavailable at first when new levels are designed for the game.

2.3.2 Gradient Boosting Classifiers

Due to the popularity of ensemble methods in classification, gradient boosting classifiers [13] are chosen for quit prediction. Only one other model (random forest) was tried and the results are similar to the gradient boosting classifier. Gradient boosting classifiers combine the predictive power of multiple weak models into a single strong learner, reducing model bias and variance. The ensemble is built in a forward stage-wise fashion where the current model corrects its predecessor model by fitting to its pseudo-residuals. Decision trees are used as the base learners. To avoid overestimation of model generalizability, hyperparameter values are kept at the default specified by scikit-learn, the python machine learning library. These consist of setting the number of estimators to 100, the maximum depth of estimators to 3, the learning rate to 0.1, using a deviance loss function, using the Friedman mean squared error criterion, and setting the subsample value at 1 for a deterministic algorithm.

2.3.3 Model Training Architecture[3]

Five-fold student-level cross-validation is used for evaluation of model performance. In this approach, students are split into folds and a single student's data is only contained in one-fold. To avoid biasing the model, feature selection is repeatedly conducted only on the training fold data. Model-based feature selection approach is used. Based on the model's fit on the training data, features are only selected to be included if their feature importance [6] (see section 3.3.2) is more than the mean of the importance of all features. The reduced-feature training data is used for model training. The performance of the trained model is evaluated on the held-out test set. The same pipeline is followed for all the five non-overlapping folds of train-test splits. Due to the skewness in the data, area under the curve (AUC) is used as the evaluation metric [16]. AUC indicates the probability that the classifier ranks a randomly chosen quit sample higher (more likely to indicate quitting) than a randomly chosen not-quit sample. The corresponding F1 value, giving the harmonic mean between precision and recall at the default threshold between quit/not quit (0.5), is also noted. Finally, precision-recall curves are used to better understand the performance of the model and to choose an appropriate probability threshold for intervention. Feature importance and partial dependence plots (section 3.3.3) are used to interpret the final model.

3. RESULTS

3.1 Level-specific Models Versus Level-agnostic Model

3.1.1 Cross-validation Results

The first analysis is aimed at choosing between level-specific and level-agnostic modelling approaches for quit prediction (section 2.3.1). For our first comparison of the model performances, only the aggregations of 49 features corresponding to student+level+visit and student+level attributes were used for training, since the level specific models cannot benefit from level-related features. We add those additional features to the level-agnostic model in a following section. Following the modeling architecture described in section 2.3.3, the five-fold student-level cross-validation results of level-specific models for the 34 unique levels in this dataset are given in Table 1. The average AUC of the level-specific models is 0.68 ($SD = 0.11$), and the average F1 value is 0.39 ($SD = 0.16$). The level-agnostic model has a cross-validated AUC of 0.75 and F1 of 0.41. The AUC of the level-agnostic model is higher than the median and mean and close to the third quartile value of the level-specific AUCs (Figure 2). The F1 value of the level-agnostic model is higher than the median and mean of level-specific F1 values. The level-specific F1 values also have high variance.

Table 1. Cross-validation results of level-specific models for the 34 levels sorted by their order in the game.

Level	#Users	%quit	AUC	F1
downhill	124	8.13	0.93	0.82
lead the ball	123	4.32	0.94	0.33
on the upswing	124	11.82	0.83	0.30
scale	124	8.70	0.92	0.32
spider web	126	15.38	0.62	0.20
sunny day	126	23.25	0.55	0.22
through the cracks	125	13.10	0.77	0.46
wavy	127	20.78	0.54	0.18
around the tree	115	29.45	0.63	0.29
chocolate factory	121	26.11	0.65	0.29
cloudy day	121	33.78	0.65	0.39

[3] The scripts for feature engineering and modelling is open at https://github.com/Shamya/Quit-Prediction-Physics-Playground.git

diving board	120	32.50	0.61	0.36
jelly beans	122	21.04	0.59	0.29
little mermaid	115	39.46	0.56	0.33
move the rocks	114	16.70	0.60	0.21
need fulcrum	126	42.41	0.55	0.40
shark	111	44.14	0.61	0.46
tricky	107	17.75	0.78	0.32
trunk slide	116	32.56	0.67	0.35
wedge	107	7.86	0.83	0.32
yippie!	123	12.66	0.69	0.28
annoying lever	107	22.41	0.68	0.37
big watermill	101	43.70	0.62	0.46
caterpillar	95	40.24	0.67	0.53
crazy seesaw	92	35.74	0.68	0.38
dolphin show	81	46.78	0.59	0.52
flower power	74	35.45	0.65	0.42
heavy blocks	72	18.75	0.61	0.23
Jar of Coins	73	36.14	0.74	0.60
roller coaster	67	45.60	0.64	0.52
stiff curtains	58	26.47	0.45	0.08
tetris	67	39.89	0.74	0.56
work it up	57	73.76	0.68	0.77
avalanche	54	28.85	0.75	0.60

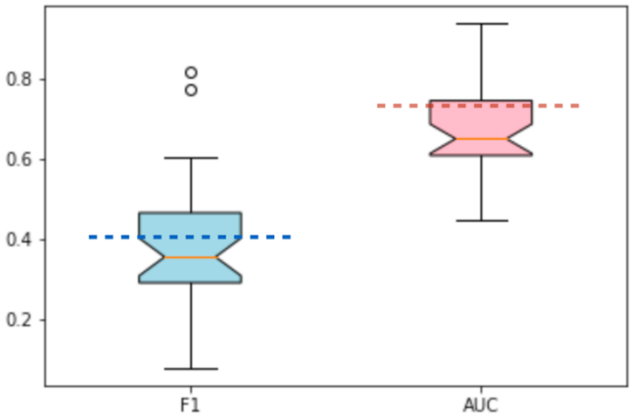


Figure 2. Box plot representing the range of AUC and F1 values of the 34 level-specific models. The box extends from the 25th to 75th percentiles, with a notch at the median. The dashed horizontal lines correspond to the values of the level-agnostic model.

3.1.2 Understanding the Model Differences

The qualitative differences between the two approaches can be explored by contrasting the features selected by each (Table 2). Feature selection for this analysis is done on the full data. The level-agnostic model seems to mainly select general features like past quits, pauses, badges, visits, level restarts, and ball resets which are common across levels. While level-specific models include these features, they also incorporate additional features related to finer-grained aspects of gameplay like the placement of pins and the drawing of specific machines (in the current 60-second time bin, in the current visit, and across visits). For instance, one of the levels named *diving board* is solved using a springboard. Among the ten features selected by this level's specific model, six of them correspond to the specific gameplay actions that one can observe a student take (e.g., total springboards drawn, total pins drawn (pins are used to hold the springboard on the screen), current number of pendulum objects on screen, and total nudges). A similar trend is seen in most level-specific models. Note that the number of level-specific models selecting any specific agent-related feature (as shown in Table 2) is distributed across agents, as most levels can be solved by only a subset of these agents.

Table 2. Comparing top features selected in level-agnostic and level-specific models.

Selected Feature	In level-agnostic model?	In how many level-specific models (out of 34)
Number of visits made by the student to this level so far	Yes	25
Total pause duration in the level so far	Yes	26
Number of past quits by the student in the level	Yes	25
Number of badges received in the level by the student so far	Yes	22
Number of restarts by the student in the level so far	Yes	20
Number of ball resets in the visit so far	Yes	23
Total ball resets in the level so far	Yes	20
Total pins drawn in the visit so far	Yes	20
Total pendulums drawn in the visit so far	Yes	17
Total nudges in the visit so far	No	32
Total nudges in the level so far	No	30
Total pins placed in the level so far	No	28
Total free form objects drawn in the visit so far	No	24
Current number of free form objects on the screen	No	25
Total ramps drawn in the level so far	No	21
Total ramps drawn in the visit so far	No	18
Total free form objects drawn in the level so far	No	17

Total pendulums drawn in the visit so far	No	15
Current number of pendulum objects on the screen	No	10

3.2 Enhancing the Level-agnostic Model

3.2.1 *Feature Additions*

Counter to the expectation that the individualized models may perform better, the AUC value of the level-agnostic model was 7 percentage points higher than the average AUC of the level-specific models. This could be attributed to the ability of the level-agnostic model to leverage the larger amount of data to identify generalizable features for quit prediction.

However, it may be possible to achieve even better predictive performance in a level-agnostic model by exploiting the level-related features (section 2.2.2 d). To examine this, we re-fit the level-agnostic model, now also incorporating the level-related features. Recall that there are two kinds of level related features – pre-defined features that can be defined for any new levels (indicating what agents and concepts are involved in solving the level) and features that use past student data to determine average behaviors for other students on the level, such as the number of objects used. We tested each type of additional features separately (Table 3, model #2 and #3). Adding just the predefined features had very little effect on the output (model #2). By contrast, incorporating the ten level-related features that use past students' data appears to improve the AUC value, though only by a modest 0.04 (model #3).

Table 3. The performance of the original level-agnostic model and various extensions to the model with level-related and student-related features.

#	Feature Set(s)	#Features	AUC	F1
1	Level-agnostic Model	44	0.75	0.41
2	Model 1 + Predefined Level-related Features	51	0.75	0.42
3	Model 2 + Level-related Features from Past Data	61	0.79	0.45
4	Model 1 + Student-related Features (level-agnostic features only)	84	0.79	0.49
5	Model 3 + Student-related Features (all features)	101	0.81	0.51

Finally, we investigated whether we can enhance the model by adding features pertaining to the student's whole history of past play (student-related features; section 2.2.2 c). We see that there is a modest improvement to the AUC values (Table 3, model #4 and #5). Note that model #4 (like model#1) doesn't contain level-related features and hence is level-agnostic. With an AUC of 0.79, model #4 could be used for new levels of the game where we do not have past student data to compute level-related features. For the current levels of the game, the best performing model (model #5) has an AUC of 0.81. Across the five folds, the AUC values of the held-out test sets have a low standard deviation of 0.01.

3.2.2 *Understanding Model Performance*

The AUC values above show that the best model (#5) is good at distinguishing students who will eventually quit from other students, but the F1 values are surprisingly low, considering the AUC. We can further understand the full model's (model #5) performance for different thresholds by examining a precision-recall (PR) curve (Figure 3) generated for all test set predictions. We see that precision is close to perfect for any threshold where recall is at or below 0.2. Additionally, recall is perfect when precision drops to 0.3. In between these extremes, the relationship between precision and recall is nearly linear, offering a clear trade-off between which of these two metrics is optimized for. Based on the characteristics of an intervention, a custom threshold on the probability can prioritize recall over precision or vice versa.

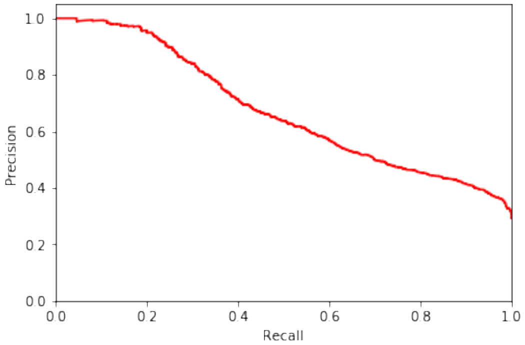


Figure 3. Precision-Recall curve of the final model (model #5).

3.3 Final Model Interpretation

3.3.1 *Selected Features*

Out of 101 features, a total of 34 features were selected by the final model (model #5). The 21 features are student-related features (out of a possible 40 student-related features), 2 are level-related features (out of a possible 17), 6 are student+level related features (out of a possible 17), and 5 are student+level+visit related features (out of a possible 27). Table 4 lists the top 15 features. Similar to the original level-agnostic model (model #1), the selected features focus on high-level game activities like visits, badges, past quits, time spent, level restarts, and experience with agents across visits and other levels. There is no student+level+visit related feature in the top 15 selected features. The final model (model #5) has 10 student-related features out of the top 15 features; note that these student-related features were not available to the original level-agnostic model. These features continually track the student's progress across all the levels.

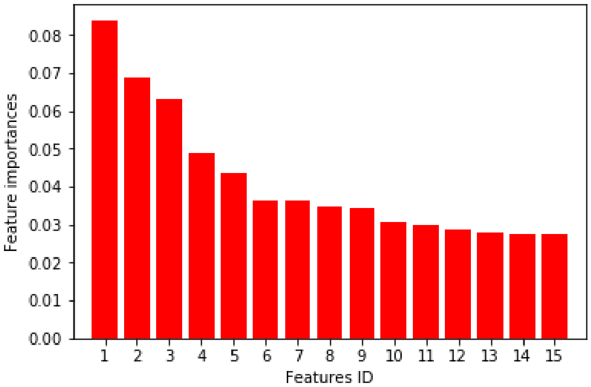


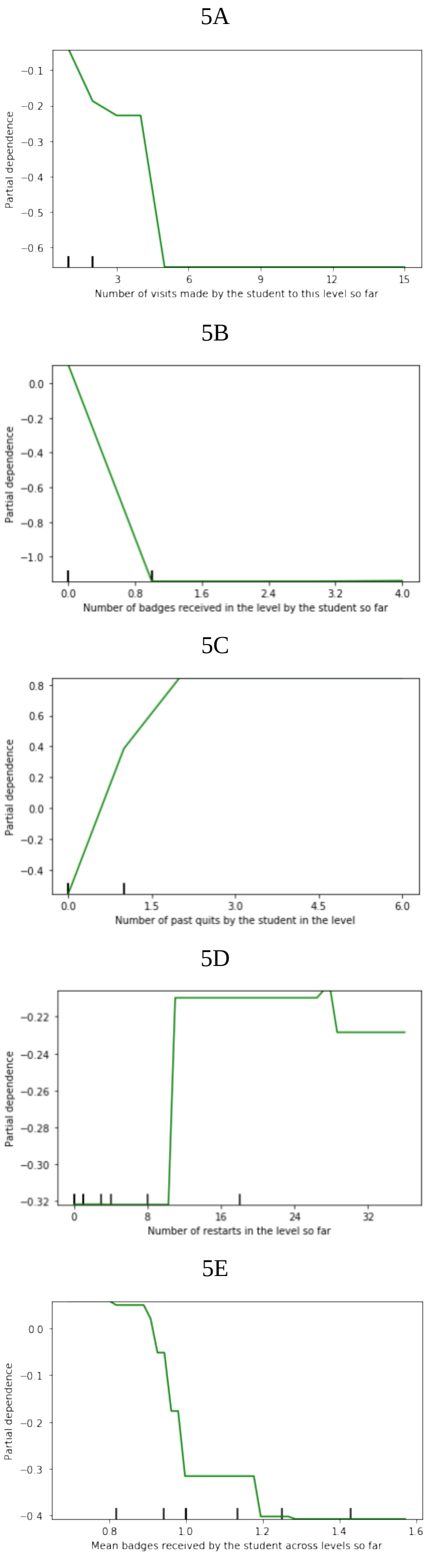
Figure 4. The feature importance of the top 15 features selected by the final model (model #5). The mapping between feature IDs and feature names is given in Table 4.

Table 4. Top 15 features selected by the final model[4] (model #5). Feature type – SLV=Student+Level+Visit, SL=Student+Level, L=Level, S=Student

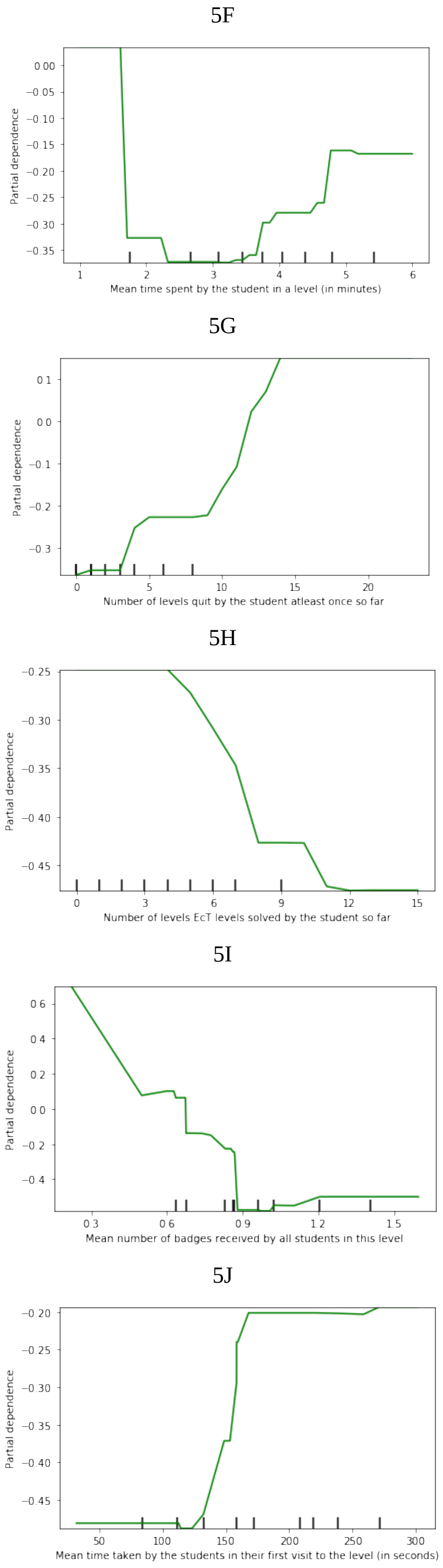
Feature ID	Selected Feature	Feature Type
1	Number of visits made by the student to this level so far	SL
2	Standard deviation of the total time spent by the student across levels so far	S
3	Mean number of badges received by all students in this level	L
4	Number of past quits by the student in the level	SL
5	Number of badges received in the level by the student so far	SL
6	Standard deviation of the total pendulums drawn by the student across levels so far	S
7	Standard deviation of total freeform objects drawn by the student across all levels so far	S
8	Mean badges received by the student across levels so far	S
9	Total pause duration in the level so far across all visits	SL
10	Mean time spent by the student in a level	S
11	Standard deviation of the number of ball resets by the student across levels so far	S
12	Standard deviation of the number of visits made by the student across levels so far	S
13	Mean pause duration of the student across levels so far	S
14	Standard deviation of badges received by the student across levels so far	S
15	Mean number of pendulums drawn by the student across levels so far	S

3.3.2 Partial Dependence Plots

Partial dependence plots (PDP) [13], originally proposed to interpret gradient boosting algorithms, have since been used with many predictive models to understand the dependence of model predictions on the covariates. Intuitively, partial dependence refers to the expected quit probability (*logit(p)*) as a function of one or more features. For example, the top right plot (5B) in Figure 5 gives the partial dependence between the mean number of badges received by students in a level (level-related feature) and the logit of quit probability after controlling for all the other features. Negative partial dependence values (y-axis) imply that for the corresponding value of the feature, it is less likely to predict quit=1. Similarly, a positive partial dependence for a feature value implies that it is more likely to predict quit=1 for that feature value. In our example, levels with mean numbers of badges earned below 0.6 are more likely to be quit by students. In general, as one might expect, there is a negative relationship between quitting and the mean number of badges received by students in a level. The higher the value of partial dependence, the stronger the relationship between the feature value and the outcome of quitting. More generally, the larger the range of the dependence value, the larger the overall influence of that feature on the model prediction.



[4] All selected features listed at - https://github.com/Shamya/Quit-Prediction-Physics-Playground.git



5F

5G

5H

5I

5J

Figure 5. Partial dependence of quit probability on some of the selected features. Note that the range of y axis is different for each plot; larger ranges indicate that the feature is more predictive overall.

Below is the summary of our interpretation of some of the features (Figure 5) selected by the final model. Most of these align with a general intuition of the game attributes and student behavior. Note that this analysis is intended only for a high-level model interpretation. The model decision-making is more complex and involves interactions between different sets of features.

1. A student who revisits a level is less likely to quit the level. This could indicate student interest in solving the level. (Figure 5A; student+level-related feature)
2. A level in which students have received fewer badges (mean < 0.6; note that a student may earn multiple badges in a level) is more likely to see quitting behavior in future students. This could indicate the inherent level difficulty. (Figure 5B; level-related feature)
3. A student who has previously solved the level is less likely to quit in their revisits to the level. This could indicate that the student generally understands the level and is trying to solve it with different agents. (Figure 5C; student+level-related feature)
4. A student who has quit a level in the past is more likely to quit the level again. This could indicate that the student is struggling with a concept or how to apply it in a way that is preventing him/her from succeeding in the level. (Figure 5D; student+level-related feature)
5. A student who has restarted a level fewer times is less likely to quit the level. Higher numbers of level restarts could indicate struggle. (Figure 5E; student+level-related feature)
6. A student who has received a higher number of badges (mean badges > 0.9) in the past levels is less likely to quit a future level. This could indicate a student who generally understands the physics concepts better. (Figure 5F; student-related feature)
7. A student who either spends under 2 minutes or over 5 minutes on average across levels is more likely to quit future levels. This feature is discussed in section 4.2. (Figure 5G; student-related feature)
8. A student who has quit more levels in the past is more likely to quit a future level. This could indicate low competence and/or disengagement. (Figure 5H; student-related feature)
9. A student who has solved more number of levels that involve the concept "energy can transfer" (EcT) is less likely to quit a level in the future. EcT is a relatively complex physics concept. In our past research [18] we have seen evidence that levels that include EcT are associated with higher student frustration. (Figure 5I; student-related feature)
10. A level in which students spend less time in average is more likely to be solved correctly by a future student. This could indicate lower level difficulty. (Figure 5J; level-related feature)

4. DISCUSSION

In this paper, we describe an automated detector we developed to predict if a student will quit a specific level they have started, within the game Physics Playground. Multiple sets of features were engineered to capture student-related, level-related and gameplay-related information over time. We compared the performance of models trained on data from single levels to the performance of a single level-agnostic model trained on the data from 34 levels. Contrary to our initial expectations, the level-agnostic model (#1 above) performed better than almost three-fourths of the level-specific models. After adding level-related features (which cannot be used in level-specific models), the resultant model (#3 above) performed better than 29 (out of 34) level-specific models. Among the five level-specific models that outperform model #3, four of them are the first four levels encountered by the students in the game and are designed to be easy. All five of the outperforming levels have around 10% of student visits ending in quitting whereas the overall incidence of quitting behavior is 28.77%. Comparing the features selected by the two kinds of models reveal the emphasis of the level-agnostic model on generalizable student behavior, while the level-specific models focus on low-level gameplay related features. The performance of the level-agnostic model is further enhanced by adding student-related features (model #4, #5 above). The final combined model (#5) selects 34 out of 101 features, which are interpreted using the feature importance scores and partial dependence plots. Due to the superior performance of the level-agnostic model and its ability to transfer to new levels and the levels with limited data, we recommend its usage over the level-specific models. Visualizing the final model with feature importance and partial dependence graphs, we find insights on which student behavior is more indicative of quitting. More analysis is needed to validate these claims.

Given the model's level of AUC, it appears to be of sufficient quality to use in intervention, identifying a student who is struggling and could benefit from learning supports before they quit the level. Our final model has a clear trade-off between precision and recall, shown in the precision-recall curve in Figure 3. Depending on the properties of a specific intervention, an appropriate threshold could be set on the classifier probability to decide whether a student is sufficiently likely to quit to justify an intervention.

4.1 Limitations

There are some potential limitations to the approach presented here. First of all, there are limitations arising from our choice to label all data in a student's visit to a level as to whether the student eventually quit. By labeling all data in the visit as quit, we may predict quitting before the behaviors have emerged that lead to quitting, and may intervene too early. This also leads to the risk of interfering with student persistence [25][11]. This risk could be mitigated by using interventions that allow the student to continue their efforts if they feel that they are not yet ready for an intervention.

Another limitation is in the generalizability of the model we have developed. Physics Playground is played by students of various age range and representing a diverse range of backgrounds, but the students in this dataset are of similar ages and live in the same region. Hence, it is important to test the generalizability of the model on data from a broader and more diverse range of students. As a next step, we are collecting data from a middle school in New York City where over 80% of students are economically disadvantaged, 97% belong to historically disadvantaged groups and all students enter the school with test scores far below proficiency. We also intend to collect data from a broader range of levels and test model applicability within this broader range of contexts.

4.2 Future Work

The goal of quit prediction is to identify student struggle in real-time to intervene meaningfully. Towards this end goal, the Physics Playground team is building an array of cognitive and affective supports that can be delivered when a student is predicted to be at risk of quitting to improve students' experience and learning. Ideally, these interventions should be based on an understanding of why a student is likely to quit, which our current model does not yet reveal. For example, a student may quit a level after putting in considerable effort, or rather quickly after minimal effort. A student may quit a level to replay other levels to achieve a gold badge, or may seek to follow a soft underbelly strategy [1], searching for a level easy enough to complete. As reported in section 3.3.3 (Figure 5F), there are two distinct quitting behaviors associated with time spent in a level. A student spending very little time in a level is more likely to quit the level. This may occur when the student is engaging in soft underbelly strategies, or when the student is putting in limited effort. Other students quit a level after considerable time and effort, indicating that they are struggling, possibly in some cases even wheel-spinning [e.g. 4]. Future work to differentiate *why* a student is likely to quit may help an intervention model to differentiate why a specific student needs support and personalize the support delivered to that student.

A learner playing a game experiences a range of emotions while engaging with the game. These can influence learning outcomes by influencing cognitive processes [12]. Knowing students' affective experience could provide deeper insights into the causes of quitting behavior. In past research [17], video-based and interaction-based affect detectors were built for Physics Playground to identify the incidence of affective states like flow, confusion, frustration, boredom, and delight. Combining quit prediction with affect detection could help us make a fuller assessment of the student experience in the learning game to provide more optimal support.

In conclusion, the key finding of this paper is that for a well-engineered set of features, a level-agnostic model of quit prediction in this learning game performs better than most level-specific models.

5. ACKNOWLEDGMENTS

The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A170376. 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] Baker, R. S., Mitrović, A., and Mathews, M. (2010, June). Detecting gaming the system in constraint-based tutors. In *International Conference on User Modeling, Adaptation, and Personalization*. pp. 267-278. Springer, Berlin, Heidelberg.

[2] Baker, R.S.J. 2007. Modeling and understanding students' off-task behavior in intelligent tutoring systems. In Proceedings of the *SIGCHI conference on Human factors in Computing Systems*. pp. 1059–1068.

[3] Baker, R.S.J., Corbett, A.T., and Koedinger, K.R. 2004. Detecting student misuse of intelligent tutoring systems. In *Intelligent Tutoring Systems*. pp. 54–76.

[4] Beck, J. E., and Gong, Y. 2013. Wheel-spinning: Students who fail to master a skill. In *International Conference on Artificial Intelligence in Education*. pp. 431-440. Springer, Berlin, Heidelberg.

[5] Bergstra, J., Casagrande, N., Erhan, D., Eck, D., and Kégl, B. 2006. Aggregate features and adaboost for music classification. In *Machine learning*. *65*(2-3), 473-484.

[6] Breiman, L., and Friedman, J. *Classification and regression trees*, 1984.

[7] Csikszentmihalyi, M. 1996. *Flow and the psychology of discovery and invention*. New York: Harper Collins.

[8] D'Mello, S., Cobian, J., and Hunter, M.: Automatic Gaze-Based Detection of Mind Wandering during Reading. In *Proceedings of the 6th International Conference on Educational Data Mining*. pp. 364–365. International Educational Data Mining Society.

[9] Daniel, S. M., Martin-Beltrán, M., Peercy, M. M., and Silverman, R. 2016. Moving Beyond Yes or No: Shifting From Over-Scaffolding to Contingent Scaffolding in Literacy Instruction With Emergent Bilingual Students. In *TESOL Journal*. *7*(2), 393-420.

[10] Dicerbo, K., and Kidwai, K. 2013. Detecting player goals from game log files. In *Educational Data Mining 2013.*

[11] Duckworth, A. L., Peterson, C., Matthews, M. D., and Kelly, D. R. 2007. Grit: perseverance and passion for long-term goals. In *Journal of personality and social psychology*, *92*(6), 1087.

[12] Fiedler, K., and Beier, S. 2014. Affect and cognitive processes in educational contexts. In R. Pekrun and L. Linnenbrink-Garcia (Eds.), *International handbook of emotions in education* (pp. 36-56). New York, NY: Routledge.

[13] Friedman, J. H. 2001. Greedy Function Approximation: A Gradient Boosting Machine. In *Annals of Statistics* 29: 1189–1232.

[14] Gee, J. P. 2003. *What digital games have to teach us about learning and literacy*. New York: Palgrave Macmillan.

[15] He, J., Bailey J., Benjamin, Rubinstein, I., and Zhang, R. 2015. Identifying at-risk students in massive open online courses. In *AAAI*.

[16] Jeni, L. A., Cohn, J. F., and De La Torre, F. 2013. Facing imbalanced data–recommendations for the use of performance metrics. In *Affective Computing and Intelligent Interaction.* ACII 2013. Humaine Association Conference on. IEEE, 245–251.

[17] Kai, S., Paquette, L., Baker, R. S., Bosch, N., D'Mello, S., Ocumpaugh, J., Shute, V., and Ventura, M. 2015. A Comparison of Video-Based and Interaction-Based Affect Detectors in Physics Playground. In *International Educational Data Mining Society*.

[18] Karumbaiah, S., Rahimi, S., Baker, R.S, Shute, V. J., and D'Mello, S. 2018. Is Student Frustration in Learning Games More Associated with Game Mechanics or Conceptual Understanding?. In *International Conference of Learning Sciences*. ICLS 2018.

[19] Ke, F. 2009. A qualitative meta-analysis of computer games as learning tools. In *Handbook of research on effective electronic gaming in education*. 1, 1-32.

[20] Lomas, D., Patel, K., Forlizzi, J. L., and Koedinger, K. R. 2013. Optimizing challenge in an educational game using large-scale design experiments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. pp. 89-98. ACM.

[21] Mills, C., Bosch, N., Graesser, A., and D'Mello, S. K. 2014. To Quit or Not to Quit: Predicting Future Behavioral Disengagement from Reading Patterns. In S. Trausan-Matu, K. Boyer, M. Crosby and K. Panourgia (Eds.), *Proceedings of the 12th International Conference on Intelligent Tutoring Systems*. ITS 2014. pp. 19-28. Switzerland: Springer International Publishing.

[22] Rowe, E., Asbell-Clarke, J., Baker, R. S., Eagle, M., Hicks, A. G., Barnes, T. M., ... and Edwards, T. 2017. Assessing implicit science learning in digital games. In *Computers in Human Behavior*. 76, 617-630.

[23] Rowe, J. P., McQuiggan, S. W., Robison, J. L., and Lester, J. C. 2009. Off-Task Behavior in Narrative-Centered Learning Environments. In *Artificial Intelligence for Education*. pp. 99-106.

[24] Sao Pedro, M., Baker, R.S.J.d., and, Gobert, J. 2012. Improving Construct Validity Yields Better Models of Systematic Inquiry, Even with Less Information. In *Proceedings of the 20th International Conference on User Modeling, Adaptation and Personalization*. UMAP 2012, 249-260.

[25] Shechtman, N., DeBarger, A., Dornsife, C., Rosier, S., and Yarnall, L. 2013. Promoting grit, tenacity, and perseverance: Critical factors for success in the 21st century. Draft released by the *US Department of Education Office of Educational Technology*.

[26] Shute, V. J., and Ke, F. 2012. Games, learning, and assessment. *In Assessment in game-based learning.* pp.43-58. Springer New York.

[27] Shute, V., and Ventura, M. 2013. *Measuring and supporting learning in games: Stealth assessment*. Cambridge, MA: The MIT Press.

[28] Squire, K. D. 2008. Video games and education: Designing learning systems for an interactive age. In *Educational Technology*. 48(2), 17.

[29] Wang, L., Kim, Y. J., and Shute, V. 2013. Gaming the system" in Newton's Playground. In *AIED 2013 Workshops Proceedings Volume 2 Scaffolding in Open-Ended Learning Environments.* OELEs. p. 85.

[30] Whitehill, J., Williams, J. J., Lopez, G., Coleman, C. A., and Reich, J. (2015). Beyond prediction: First steps toward automatic intervention in MOOC student stopout. In *Social Science Research Network*.

[31] Yang, D., Sinha, T., Adamson, D., and Rose, C. P. 2014. "Turn on, tune in, drop out": Anticipating student dropouts in massive open online courses. *In NIPS Workshop on Data-Driven Education*.

Who they are and what they want: Understanding the reasons for MOOC enrollment

R. Wes Crues
University of Illinois at Urbana-Champaign
crues2@illinois.edu

Nigel Bosch
University of Illinois at Urbana-Champaign
pnb@illinois.edu

Carolyn J. Anderson
University of Illinois at Urbana-Champaign
cja@illinois.edu

Michelle Perry
University of Illinois at Urbana-Champaign
mperry@illinois.edu

Suma Bhat
University of Illinois at Urbana-Champaign
spbhat2@illinois.edu

Najmuddin Shaik
University of Illinois at Urbana-Champaign
shaik@illinois.edu

ABSTRACT

The diversity in reasons that students have for enrolling in massive open online courses (MOOCs) is an often-overlooked aspect while modeling learners' behaviors in MOOCs. Using survey data from 11,202 students in five MOOCs spanning different academic disciplines, this study evaluates the reasons that students enrolled in MOOCs, using an unsupervised learning method, Latent Dirichlet Allocation (LDA). After fitting an LDA model, we used correspondence analysis to understand whether these reasons were general, and could be invoked across the five MOOCs, or whether the reasons were course-specific. Furthermore, log-linear models were employed to understand the relations between the reasons students enrolled, the course they took, and their background characteristics. We found that students enrolled for many different reasons, and that their age was statistically related to the reasons they gave for taking a MOOC, but their gender was not. The paper concludes with a discussion of how instructors and course designers can use this information when creating new—or redesigning existing—MOOCs.

Keywords

MOOCs, informal education, text mining

1. INTRODUCTION

Massive open online courses (MOOCs) have been celebrated because they offer education to wide groups of students who may not otherwise have access to their rich content; they provide access to well-respected experts; they have a relatively low cost; and they are convenient. On the other hand, MOOCs have been criticized because they have high attrition and low completion rates. We acknowledge that there are high attrition and low completion rates, but if students sign up with the intent of only learning some aspects of what is offered in the MOOC, and not necessarily with the intent to learn everything that the MOOC has to offer, this ought not to be considered a failure. Acknowledging that MOOC learners have different reasons for enrolling in MOOCs—for example, to improve their skills, gain access to new knowledge, or dabble in an area they find intriguing—we examine whether the reasons students offer are MOOC-specific or content-generic for five MOOCs. We do this with the intent to distinguish whether the reasons that learners have for enrolling in MOOCs is linked to their background (age or gender) or to the specific course they have enrolled in. By finding ways to classify these reasons reliably, we will be in position to understand the relation between why students enroll in these courses and how successfully they navigate the course.

Although it may be advantageous for students to participate in all aspects of a MOOC and to complete the course, MOOCs are beginning to accommodate different paths and different outcomes. For example, Coursera[1] (one of the most popular MOOC providers) offers verified course completion certificates for students who wish to obtain proof of their accomplishments, but also allows students to enroll for no credit and sample whatever course materials they wish. However, most MOOCs do little to support the multiplicity of learning objectives that students may have for taking a particular MOOC. Understanding students' reasons for enrolling in a MOOC could put instructors in the position to make accommodations, potentially improving students' learning experiences.

A growing body of literature has investigated why students enroll in MOOCs (e.g., [7, 3, 5, 16, 23, 27]). These studies have used survey methods with closed-form responses or have used interviews. With surveys using closed-form responses, students are forced to select from a list of reasons; and with interviews, typically, only a limited number of students may be reached. In the current study, we investigated more than 11,000 students enrolled in five MOOCs, across several disciplines, using Latent Dirichlet Allocation (LDA) [2] to analyze their responses to an open-ended survey. We then used the probabilities from the LDA model to assign

[1] https://www.coursera.org

each student to one of the topics (i.e., reasons for enrolling) generated by the LDA model. After we found the most probable reason a student enrolled, we cross-classified students by their most probable topic (i.e., reason) and the course for which they enrolled. We then visualized these relationships using correspondence analysis.

In this paper, we contribute to understanding student behavior in MOOCs by examining the reasons that students offered for enrolling in MOOCs, and the extent to which these reasons are unique to the specific MOOCs or whether they apply more generally, across MOOCs. Additionally, we advance understanding by using LDA and the results of log-linear models to hone in on specific relationships between student background characteristics and their reasons for enrolling. Using these results, we conjecture about how instructors and course designers could use this information to improve their courses and their students' learning experiences, thus contributing to the discussion about improving instruction for diverse learners.

2. RELATED WORK

Several studies have sought to make sense of what kinds of students enroll in MOOCs, and why. Specifically, these studies have examined students' background characteristics and why they take MOOCs. We discuss some of these works in the following subsections.

2.1 Goals for Enrollment in MOOCs

Current findings on why students enroll in MOOCs have revealed that students enroll in these courses for many different reasons. Hew and Cheung [11] identified common trends for why students enrolled in MOOCs, including: (1) a general interest in learning; (2) a desire to receive formal recognition of their knowledge; (3) an intent to explore course content without a strong desire to receive such recognition; and, (4) an interest or general curiosity in taking a MOOC. Next, we explore some of these themes in more depth.

Zheng et al. [27] interviewed students who took MOOCs and asked about their reasons for enrolling in MOOCs. Some students in their study were fulfilling their current needs, such as supplementing a for-credit course, or to help with their current position, either as students or in a workplace setting. Other students offered that they took the course to develop a social connection with others who shared similar interests. Additionally, they found some who enrolled did so to prepare for future job opportunities or to gain experience in a field they might study in a more formal manner after taking the MOOC. Finally, some of the students in this investigation enrolled in the MOOCs because they were interested in satisfying (broadly) their curiosity. Along these lines, it has been posited that MOOCs function as previews of what might be offered to students in a for-credit university course [15].

Kizilcec and Schneider [16] developed the Online Learning Enrollment Intentions (OLEI) questionnaire, which asked students to select whether or not each of 13 different reasons for enrolling in a MOOC applied to them. These reasons included career-related interests, formal education, social opportunities, potential career benefits, personal enrichment, and prestige. Liu, Kang, and McKelroy [18] found most of the students in a set of MOOCs took those MOOCs for personal interest, or to improve their current knowledge of the job and prepare for future job prospects. To this end, the subject matter of the course was also indicative of the reason a student might take a MOOC. For example, Kizilcec and Schneider [16] found that students in a humanities course might have taken the course out of curiosity, versus students in a social science or health-care-related course, who might have taken the course for career benefits [5].

Others have investigated whether students' reasons for enrolling in a MOOC impacted their behavior during the course and whether or not students completed the course. For example, de Barba et al. [7] found that students' motivation and their interests were related to how they engaged with the course's quizzes and videos. They also investigated how motivation—either intrinsic motivation or situational interest—was related to a student's final grade in an introductory economics MOOC. Others, however, observed no relation between student motivation and the grades earned in MOOCs [3]. On the other hand, Pursel and co-authors [23] found that students who had the intention to be an active participant in a MOOC had higher odds of completing the MOOC. In other words, those who stated they were motivated to finish the MOOC were actually more likely to do so.

We also note that a few studies have investigated students' reasons for enrolling in MOOCs by analyzing open-ended survey questions. For example, Robinson and colleagues [25] analyzed n-grams from the responses to a survey question that asked students how the course material was useful and how they planned to use the knowledge gained from the course. Using regularized regression, they found students whose answers included words that indicated a plan to readily apply the knowledge gained from the course, and expected to use the skills learned from the course in a vocational setting, were more likely to earn a certificate than students whose responses indicated an interest in obtaining formal recognition. In another investigation of open-ended survey responses, Crues et al. [6] found that students' reasons for enrolling in a MOOC clustered into four interpretable reasons, and some of the reasons were related to actively engaging in portions of the course; however, these reasons were not statistically related to remaining engaged in the course overall. In general, much more can be learned from students' motivations and goals for enrolling in MOOCs, and this new knowledge can be utilized to further an understanding of students who take these courses.

2.2 Role of Gender and Age in MOOCs

MOOCs can provide informal experiences for students, with few barriers and no requirements for enrollment, but this also leaves MOOCs without traditional educational data about student background characteristics. However, there have been several studies that have explored the relations among student characteristics, enrollment patterns, and behavior in MOOCs. In this paper, we focus on the relation between two background characteristics—gender and age—in understanding reasons for enrolling and behavior in MOOCs. We have chosen to examine gender because of MOOCs' great promise to offer educational experiences to all, which has particular importance for women, who oftentimes have fewer educational opportunities than men. In addition, men and women might have different patterns of enrollment in different courses, and having this information could be vital for modifying and improving a course. We have also chosen to investigate age because older learners and younger learners might engage with MOOCs for very different reasons, and we want to document evidence on this issue.

With respect to gender, differences have been observed in whether males or females take a certain MOOC. Specifically, courses focused on science technology, engineering, and mathematics (STEM) tend to be dominated by male students [24, 10, 3]. For example, Breslow and co-authors [3] investigated "Circuits and Electronics" and found that 88% of the students who submitted an end-of-the-course survey were male. Women, more than men, are more numerous in other fields [20, 24]. And although men are more numerous in some STEM fields, medicine seems to be an exception: a course in medicine analyzed by Kizilced and Schneider [16] was overwhelmingly female—91% of students were female. We suspect that knowing the gender composition of the course is useful information to the instructor, especially if an instructor's goal is to attract more women or more men to the course.

The age of students in MOOCs has often revealed that students are young [5], with little variation between courses in different academic disciplines [16, 20]. Others, however, have found there to be a wide range of ages in classes (e.g., [3]), and that age varies based on geography [10]. The disparate findings on the age of MOOC students suggests that the relation between student age and participation in a MOOC is still murky and further research could be done to clarify this relationship.

Students' ages and genders have often been found to share (at best) a weak relationship with their reasons for enrolling in a MOOC. With respect to gender, Crues and colleagues [6] observed that students' reasons for enrolling in a computer science MOOC and gender did not share a significant statistical relationship.

Some have reported that females selected more reasons for enrolling in a MOOC on the Online Learning Enrollment Intentions (OLEI) scale than males [16]. In that study, reasons for enrolling in a MOOC were found not to be related to the age of a student. However, students who were using the MOOC to supplement their formal schooling were generally younger than students who did not indicate this reason for enrolling in the MOOC [16].

Although student gender and age have been found not to share a relationship with student reasons for enrolling in MOOCs, these background characteristics have been identified as sharing a relationship with student behaviors in MOOCs. For example, female students tend to spend more time viewing videos and completing assignments than males [24]. Although Swinnerton, Hotchkiss, and Morris [26] found that gender was not statistically related to the number of comments a student posted in a MOOC forum, others [24] found that females in non-science courses posted more inquiries in forums than males, but the opposite has been found to be true for science courses. Furthermore, it has been found that the reasons students gave for enrolling in a MOOC were related to their forum participation—men who enrolled to complement their career goals and women who did so to explore the content (e.g., they were curious about the course's subject matter) were more active in the forums than students who gave other reasons for taking the MOOC [6]. Findings have been inconclusive on whether gender shares a relationship with completing a MOOC: some investigations have found that gender shares a relationship with remaining persistent in a MOOC (e.g., [6]) or earning a certificate, depending on the course (e.g., [24]), while others have not observed this effect (e.g., [3, 20]).

Students' age has also been used to shed light on students' behavior in MOOCs. It was found that older students were more engaged with a MOOC than younger students; older students were found to have accessed digital course materials more frequently than younger students [10]; and older students were more active in the course forums than younger students [26, 10]. More generally, older students have been found to access more of the course materials than younger students [20, 10]. Similar to gender, there has been inconclusive evidence about whether age shares a relationship with success and completing a MOOC. For example, some have found that age was statistically related to grades (e.g., [10]) but others have not observed this effect (e.g., [3]. Still others have found that gender and completing a MOOC are not related [20].

In general, the literature has pointed to age and gender to be of interest in predicting enrollment and success in MOOCs, but the findings are not clear. Furthermore, we need to know more about why certain students enroll in some courses, and which of these reasons apply to MOOCs, in general, and which of these reasons only apply to particular MOOCS. Gaining insight on these issues is crucial for instructors and course designers to consider for attempting to improve courses. Thus, we conducted our investigation to provide more clarity on these issues.

3. METHOD

We used survey data to understand why students enrolled in one of five MOOCs offered on Coursera: Creative, Serious, and Playful Science of Android Apps (Android), Introductory Organic Chemistry (Ochem), Subsistence Marketplaces (Subsistence), Introduction to Sustainability (Sustainability), and E-Learning Ecologies (Elearning). Students who enrolled in these courses were asked to submit a survey that asked about their background and expectations for the course, along with their age range and gender. The survey posed the questions, "Why are you taking this course? What do you hope to get out of it?" Students were able to enter an answer to both questions in one open-ended response. We call this the *reason* the student enrolled in the MOOC. We analyzed the responses to this survey to understand (1) why students enrolled in these MOOCs, (2) whether these reasons were related to specific courses or to the five MOOCs, in general, and (3) how reasons and courses were related to the students' background characteristics (gender and age).

Of the $N = 341523$ students enrolled in these MOOCs, $n = 37178$ responded to portions of the aforementioned survey; however, only $n = 12407$ students provided a reason that they enrolled in the course. As a result, these are the only students we will consider for analysis. In addition, because we used LDA to analyze the reasons that students enrolled, we removed non-English responses (using the `textcat` package in R [8]). This resulted in the total number of responses to be analyzed as $n = 11202$. The students who provided responses in English were spread throughout the five courses as shown in Table 1. The gender and age distribution for these courses is also displayed in Table 1.[2]

After we removed non-English responses, we prepared the text for analysis using the `tm` package in R [19]. Before we did any text pre-processing, there were 11058 unique words in the set of English responses. We removed stop words, punctuation, and numbers, while also transforming all characters to lower case and stemmed the terms using the Porter stemming algorithm [22]. Additionally, the term frequency-inverse document frequency (tf-idf) scores were computed for the collection of reasons. We removed terms that had tf-idf scores at or below the tenth percentile, because these terms might include more noise in the text data. After completing these pre-processing steps, we had 9952 unique terms in the set used to model these responses.

To model these responses, we used Latent Dirichlet Allocation (LDA) [2], which is a type of unsupervised topic model. Topic models are probabilistic models, which assume that a collection of documents follow an underlying latent distribution [2, 12]. LDA is a well-suited method for this problem because the reasons students gave do not have a label attached to them, and our goal was to explore the relations between reasons and MOOCs. Specifically, the LDA model is defined as

$$p(\theta, \mathbf{t}, \mathbf{w}|\alpha, \beta) = p(\theta|\alpha) \prod_{i=1}^{I} p(t_i|\theta) \cdot p(w_i|t_i, \beta), \qquad (1)$$

where θ is the topic mixture, t is the number of topics I an LDA model assumes, w is the collection of words used to fit the model, α is a vector of length t, and β is a matrix of word probabilities [2]. To estimate these models, various estimation strategies have been proposed. One approach is variational expectation maximization (VEM) [2]; however, the starting values of the algorithm are non-trivial which could result in finding local, versus global, maximums [9, 2, 13]. To combat this problem, Gibbs sampling has been proposed to estimate the unknown parameters for LDA, and identifies these parameters faster than other algorithms [9]. Before estimating an LDA model, however, one must specify the number of topics, t.

To determine the number of topics in the collection of reasons, we used the strategy proposed by Griffiths and Steyvers [9], which was implemented using the `ldatuning` package in R [21]. After estimating LDA models where the number of topics was $I = \{10, 11, 12, ..., 35\}$, we found 26 topics was close to the maximum of the metric proposed in [9]; thus we fit an LDA model with 26 topics. We show the metric's behavior versus a subset of the number of fitted topics in Figure 1.

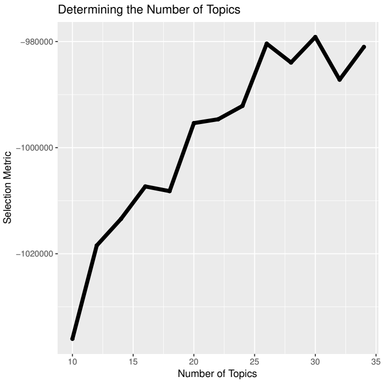


Figure 1: Plot of model fit metric in [9] versus the number of topics. When the number of topics is 26, the metric is near the maximum.

Once we determined $t = 26$, the LDA model was fit using the `topicmodels` package in R [14], where we used Gibbs sampling with 500 random starts and 5000 iterations, and the first 1000 iterations were discarded for burn-in. The final model selected was the one that had the highest posterior likelihood, and then we assigned each student to one of the 26 topics. This was done by computing the posterior probability from the LDA model, and each student's reason was assigned to whichever topic had the highest probability.

After we assigned each student to one of the topics, we cross-classified students by the topic to which they were assigned from the LDA model and the course in which they were enrolled. To test whether there was a statistical relationship between the topics and the particular course they were enrolled in, we used the χ^2 test of independence. We do not offer direct interpretations of the topics because it is difficult for humans to identify topics from a given set of terms from a topic model [4]. However, the potential relationship between courses and reasons lends itself to correspondence analysis, so we further analyzed this two-way table by correspondence analysis using `FactoMinR` [17].

Correspondence analysis was used to represent the association between reasons and courses using the data in Table 2. Plots of the estimated scores for the topics (rows) and courses (columns) represent the dependency in the table. The method can determine which reasons differentiate or are unique to particular courses and which reasons do not distinguish between the courses (i.e., the reason is common to all courses).

To investigate whether the relationships between topic and course was mediated by background characteristics, specifically gender and age, we fit log-linear models (using maximum likelihood estimation) to three-way contingency tables of topic by course by background characteristic. Our modeling strategy started with a complex model, and then we sought to find the most parsimonious model that yielded a good representation of the data. Specifically, we started by

[2] Students were able to identify as male, female, or neither of these. After filtering out students who did not provide a reason for enrolling or an answer in English, all remaining students identified as either male or female.

Table 1: Distribution of students enrolled in the five MOOCs by gender and age.

Course	Students		Gender		Age Group						
	Total	Complete Survey	Males	Females	≤ 17	18-24	25-29	30-39	40-49	50-59	≥ 60
Android	189334	4656	3589	1067	112	1055	980	1234	684	393	198
Ochem	38526	784	440	344	12	193	185	193	84	72	45
Subsistence	23854	729	312	417	3	103	161	196	97	91	78
Sustainability	76886	4199	1889	2310	17	520	917	1116	641	527	461
Elearning	12923	834	357	477	1	24	74	224	239	185	87

modeling the relationship using

$$\log \mu_{ijk} = \lambda + \lambda_i^b + \lambda_j^c + \lambda_k^t + \lambda_{ij}^{bc} + \lambda_{ik}^{bt} + \lambda_{jk}^{ct} + \lambda_{ijk}^{bct}, \quad (2)$$

where i corresponds to the levels of the background characteristics b (i.e., male and female for gender, or the 7 age groups), j corresponds to the courses, c, and k corresponds to the most probable topic, t, from the LDA model. Note then that μ_{ijk} is the number of students in cell ijk in the three-way contingency table. The analyses for gender and age were carried out separately. Once a model was chosen, we further studied the nature of the associations found in the data.

When using log-linear models, a Poisson distribution is typically assumed for the distribution of counts; however, we suspect that there was more heterogeneity within combinations of topic, course, and background than is predicted by a Poisson distribution (i.e., the data exhibit "over-dispersion"). To deal with this we used a negative binomial distribution in our log-linear models. Our conjecture that data were over-dispersed was confirmed. The dispersion parameter was large relative to its standard error and the negative binomial models yielded much better goodness-of-fit statistics. In all models and further analyses, we report the log-linear model and test statistics using a negative binomial distribution.

4. RESULTS

We first note that there were differences in student background characteristics across these five courses. From Table 1, we can see that there were more males in Android and Ochem, but more females in Subsistence, Sustainability, and Elearning. In general, there were few students aged 17 or younger in these courses. Most students were in the middle age groups. We used a likelihood ratio statistic of independence assuming a negative binomial distribution to test whether age and gender shared a statistical relationship, without respect to courses. The marginal relationship between gender and course was statistically significant (X^2 = 10.03, *df* = 4, p = .03), and the relationship between age and course was also statistically significant (i.e., X^2 = 38.49, *df* = 24, p = .03). Thus, we have evidence to believe that age and gender are statistically dependent with respect to who enrolls in these courses.

Table 2 defines the general topic model, where the five most probable words in each topic are listed with each topic and the number of student responses for each topic are displayed for each course. Note that the topics are ordered in an arbitrary manner.

To test whether there was a significant association between being enrolled in a specific course and assignment to a specific topic, we used a X^2 test of independence. Unsurprisingly, this test revealed a dependent relationship between topic and course (i.e., X^2 = 12570, *df* = 100, $p-$value $< .001$).

Furthermore, to gain insight into the nature of the relationship between topic and course, we performed a correspondence analysis. The first two dimensions account for 68.91% of the total inertia, which is a measure of the amount of association in the data (i.e., how much the data deviate from expectations under independence). The category scale values from the first two dimensions of the correspondence analysis are plotted in Figure 2. Greater distances between points for the courses indicates that there are greater differences in their profiles, with respect to the topics (a profile corresponds to the conditional distribution of topics, given course). Likewise, greater distances between points for the topics indicate greater differences in the profiles with respect to the courses.

The course points for Subsistence, Sustainability, and Elearning are close together, which indicates that these three courses have similar profiles with respect to the topics. These three courses are the least distinguishable in terms of the topics. The Android and Ochem points are far from each other and far from the other three courses, which indicates that these courses have considerably different profiles with respect to the topics and are quite distinct.

Although the absolute distances between the course and topics points are not meaningful, the relative distances between course and topic points are meaningful. For example, the points for topics (the reasons) 9, 19, and 20 are relatively close to Android, which means that these topics were given as a reason for taking Android more often than would be expected if topics and courses were independent. As can be seen in Figure 2, as we just noted, topics 9, 19, and 20 are relatively close to Android (most probable words: android, program, learn, develop, app), topic 2 is relatively close to Ochem (most probable words: chemistri, organ), topics 1, 10, 11, 15, and 17 are relatively close to Elearning (most probable words: understand, better, teach, onlin, world, way, work, current, interest, subject), topics 10, 17, and 25 are relatively close to Sustainability (most probable words: teach, onlin, interest, subject, studi, field), topics 7, 12, and 23, are relatively close to subsistence (most probable words: sustain, environment, market, social, can, chang), topics 7, 22, 23, 26 are all relatively close to Subsistence and Sustainability (most probable words: sustain, environment, sustain, system, can, chang, sustain, sustainability), and topics 10,

Table 2: Number of students matching each topic in the topic model, with distinctive words characterizing each topic.

Topic	Most Frequent 5 Words	Android	Ochem	Subsistence	Sustainability	Elearning
1	understand,better,hope,abl,gain	190	28	69	340	69
2	chemistri,organ,school,take,chemistry	76	466	6	117	20
3	take,course,the,also,reason	169	16	26	171	36
4	one,know,think,need,realli	164	24	18	234	28
5	use,make,can,like,idea	321	6	19	96	37
6	will,help,hope,give,think	182	25	24	178	30
7	sustain,environment,issu,sustainability,topic	40	2	23	406	14
8	knowledg,improv,skill,field,knowledge	255	25	43	277	49
9	android,program,app,apps,comput	1029	1	4	9	5
10	teach,onlin,educ,elearn,technolog	67	11	12	102	280
11	world,way,can,find,peopl	79	8	37	157	14
12	market,social,develop,work,countri	43	1	244	100	12
13	want,learn,know,just,curious	185	14	20	138	15
14	time,coursera,class,enjoy,great	84	52	11	170	18
15	work,current,project,area,compani	74	5	27	154	33
16	learn,new,someth,want,thing	210	8	17	111	32
17	interest,subject,area,view,point	78	6	35	219	25
18	like,interest,look,topic,see	123	5	19	112	17
19	learn,develop,want,development,basic	221	7	7	43	20
20	android,app,develop,creat,mobil	828	1	2	4	0
21	get,hope,job,good,field	85	8	7	90	14
22	sustain,system,food,product,energi	12	6	16	225	4
23	can,chang,sustain,futur,human	7	2	12	292	15
24	year,time,ive,now,tri	93	32	13	83	9
25	studi,field,degre,research,master	24	23	15	175	32
26	sustain,sustainability,concept,practic,need	17	2	3	196	6

11, 15, 17, and 25 are relatively close to sustainability and Elearning (most probable words: teach, onlin, world, way, work, current, interest, subject, studi, field). The topics in the center of the figure (i.e., 3, 4, 6, 8, 13, 14, 16, 21, and 24) are those that do not differentiate the courses and are given as reasons for all courses (probable terms include take, course, one, know, will, help, knowledg, improv, want, learn, time, coursera, learn, new, get, hope, year, time). Next, we consider how the student background characteristics are related to the reasons and the courses.

4.1 Gender, Reasons, and Courses

We fit log-linear models to understand the relationship between student gender, the topic a student was assigned to from the LDA model, and the course they took. The homogeneous association model (all 2-way interactions, but not the 3-way interaction from Equation 2) yielded an excellent representation of the data (i.e., the likelihood ratio goodness-of-fit statistic was $X^2 = 49.186$, $df = 100$, $p = .99$). Among the three possible conditional independence models (i.e., only two two-way interaction in equation 2) fit to the data, only the model where topic and gender are independent given course gave a good representation of the data (i.e., $X^2 = 16.318$, $df = 25$, $p = .91$).

Given that the topic and gender were conditionally independent given course, we could collapse over gender to study the topic by course relationship and collapse over topic to study the relationship between gender and courses [1]. We have already described the relationship between course and topic based on the correspondence analysis. Figure 2 described both males and females; in other words, there are no difference between males and females in terms of the dependency between courses and topics.

To study gender by course dependency, we refer to the middle of Table 1. We found, using a negative binomial distribution, that gender and course were dependent. Table 3 contains Haberman residuals from the independence model. We chose to use Haberman residuals, which are related to standardized Pearson residuals, because Haberman residuals are distributed $N(0, 1)$, whereas the distribution of Pearson residuals is $N(0, < 1)$ [1]. The Android course was the only course were there was a noticeable difference between males and females. The males enrolled in the Android course more than expected and females enrolled far less than expected.

4.2 Age, Reasons, and Courses

Similar to the analysis for gender, we fit log-linear models (again, using the negative binomial distribution) to the topic-by-course-by-age, 3-way table. As before, the homogeneous association model yielded an excellent representation of the data (goodness-of-fit likelihood ratio test statistic $X^2 = 470.355$, $df = 600$, $p = .99$). None of the conditional independence models yielded an acceptable goodness of fit to the data. We were not able to collapse over any of the variables to describe the association between pairs of variables [1], so we further examined the partial tables (i.e., the relationship between age and topic, age and course, and course and topic) to describe the association between pairs of variables with an emphasis on the topic-by-course interaction.

To further explore the relationship between age group, the topic from the LDA model, and the course a student took, we

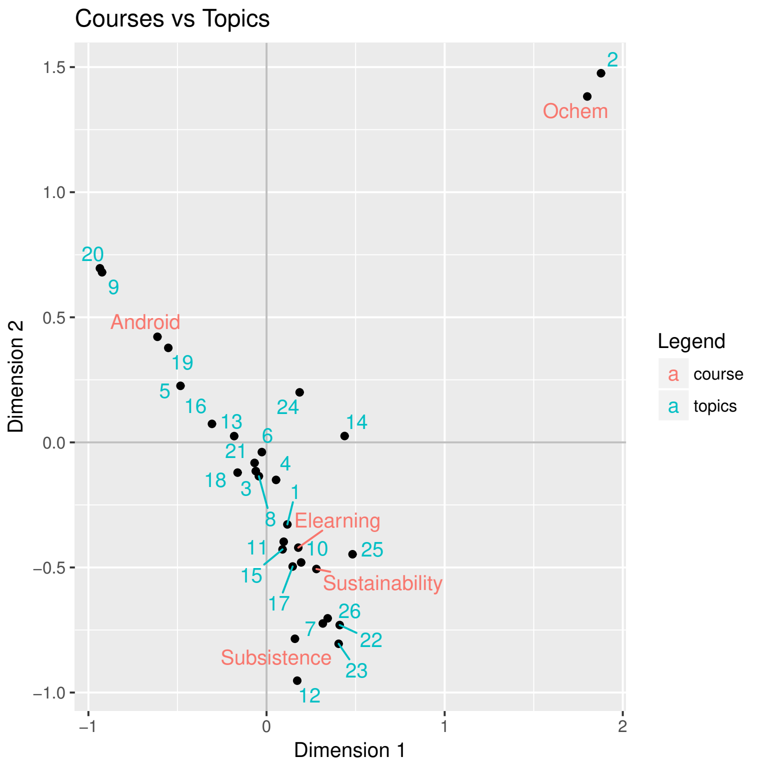


Figure 2: Topics and courses from Table 2 projected in the first two dimensions of correspondence analysis (association between topics and reasons for enrolling).

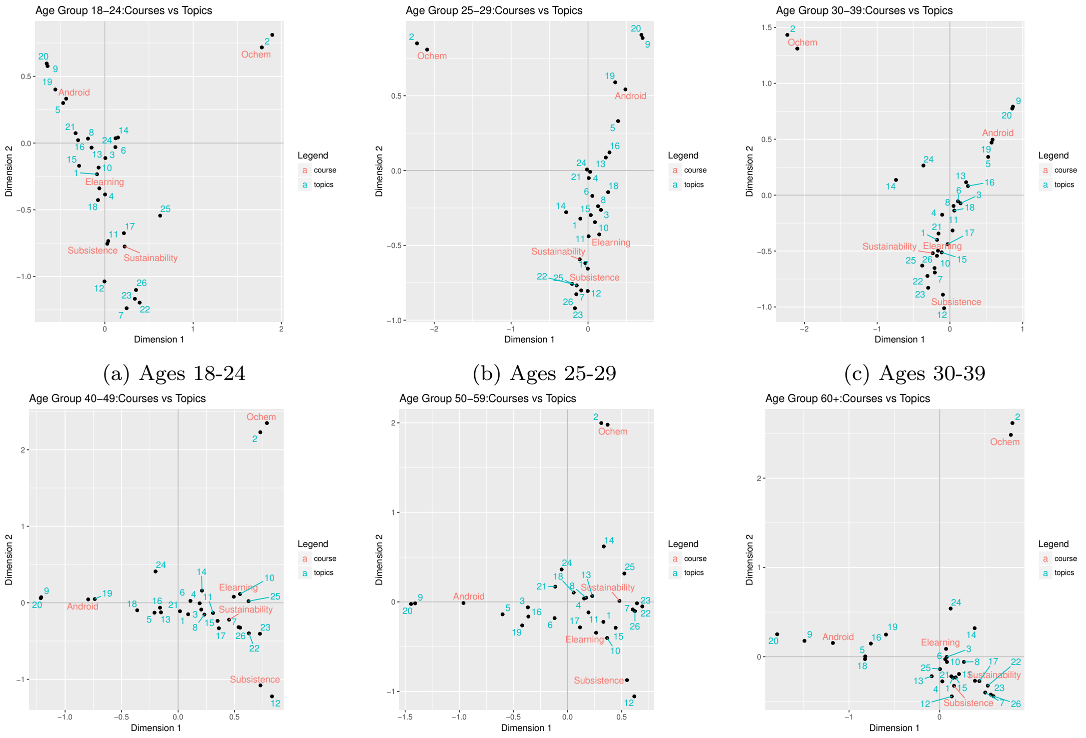
Table 3: Haberman residuals from independence using the Negative Binomial distribution.

	Course				
Gender	Android	Elearning	Ochem	Subsistence	Sustainability
Male	-2.8035	1.13055	-0.3422	1.12843	0.90563
Female	2.80349	-1.1305	0.3422	-1.1284	-0.9056

used correspondence analysis where we completed a separate analysis for each age group. We include six correspondence analysis plots for the first two dimensions in Figure 3 for all age groups except the youngest students, because there were very few students in this category ($n = 145$). Table 4 gives the proportion of total inertia accounted for by the first two dimensions of the plots in Figure 3.

Generally, Ochem is far from the other courses and, relative to the other courses, is far from all but one topic (topic 2, where the most probable words are chemistri and organ). Likewise, Android is relatively far from other courses as shown in Figure 3. In all of the plots, we observe that topics 9 and 20 are quite close to Android, which is intuitive given that the most probable words for these topics are android, program, app, and develop. Furthermore, across the different age groups, topic 19 is relatively close to Android, where the most probable words are learn and develop. Across all of the age groups in Figure 3, topic 11 is generally close to Subsistence, where the most probable words are market, social, and develop. We see that for most students, topic 10 is quite close to Elearning. The most probable words for this topic are teach, onlin, educ. In most of the plots in Figure 3, topic 17 is generally close to Sustainability, and the most probable words for this topic are interest, subject, and area. For the other topics, it is more difficult to establish a clear pattern across the different age groups. In other words, many of the topics do not consistently differentiate the courses from one another, and thus, are reasons given for all of the courses.

To further understand the relationship between students' age and the topic they were assigned to, given the course they took, we considered the Haberman residuals of the partial tables. That is, we considered the residuals for five 2-way tables, where each table corresponded to one of the MOOCs, and the rows and columns corresponded to the topics and age groups. Because Haberman residuals follow the standard normal distribution, any residual with an absolute value of two or greater is of note. Out of the 910 residuals, there were eight residuals with an absolute value greater than 2 for Android, 13 for Elearning, 8 for Ochem, 12 for Subsistence, and 4 for Sustainability. The large residuals in this case were generally for the two youngest age groups (i.e., students 24 years old and younger) or the two oldest age groups (i.e., students 50 and older). This suggests that, given the course a student took, we saw students in these four age groups were assigned to topics much more or much less than expected. This means some younger and older stu-



(a) Ages 18-24

(b) Ages 25-29

(c) Ages 30-39

(d) Ages 40-49

(e) Ages 50-59

(f) Ages ≥ 60

Figure 3: Correspondence analysis plots for all age groups except the youngest.

Table 4: Inertia accounted for by the first two dimensions, as shown in Figure 3.

Age Group	Inertia from first two dimensions
18-24	74.50%
25-29	75.89%
30-39	69.43%
40-49	63.55%
50-59	63.54%
≥ 60	65.60%

dents take courses for reasons that were not expected, when we account for the course they took.

Additionally, we considered the partial tables to understand the relationship between age group and the course a student took, given the topic they were assigned to from the LDA model. We examined the Haberman residuals for 26 tables—one for each topic; in this case, the rows and columns correspond to the age group and course a student took, and the cells contain the Haberman residuals. We found 45 out of 910 residuals with an absolute value of two or greater. Many of these larger residuals were for students in the two youngest and two oldest age groups. This suggests that students in these age groups took some of the courses much more or much less than expected, given the topic they were assigned to from the LDA model.

5. DISCUSSION

This investigation explored the reasons students gave for enrolling in one of five different MOOCs, and how these reasons related to the course students took, their gender, and their age. The five courses considered in this paper are from diverse academic disciplines and attract different groups of students.

Unlike some previous studies that have explored student goals for enrolling in MOOCs by asking them to select a reason from predetermined answer choices, students in this study specified their reasons for enrolling via an open-ended response. This afforded students the opportunity to provide more genuine responses, versus being forced to conform to a set of choices on a survey. As a result, we found 26 reasons students gave for enrolling in these MOOCs when using LDA. The number of topics for the LDA model, which must be specified, was derived empirically from the approach given in [9]. From this topic model, we observed that some students decided to enroll in a course for very specific reasons and we suspect that these specific reasons were related to the course content. This follows from the fact that some topics were very close to courses in the correspondence analysis; further support for this comes from the most probable words from each of these topics. On the other hand, some topics from the LDA model applied to all courses. These topics were those that were towards the center of the correspondence analysis plots. When examining the most probable terms for these topics, we found very general terms that did not have an apparent relationship to one of the five courses we considered.

We also examined whether students' gender or age were related to the courses they took and the reason they enrolled in the course. We first considered whether a students' gender, the course they took, and the topic they were assigned to from the LDA model were statistically related. Our analyses revealed there was not a 3-way interaction between these factors; however, our findings led us to analyze the relationship closely between topics and courses, and courses and gender. It was observed that gender did not mediate the relationship between topics and courses, thus, our findings about how the topics and courses are related is not different for males versus females. This finding is consistent with previous studies, which have found that, generally, the reason a student enrolls in a MOOC and their gender are not related (cf. [6], [16]). On the other hand, we found that there was a relationship between the courses students took and their gender. Some of the courses, such as those in the sciences, had more males, and those not in the sciences had more females. This finding parallels the enrollment patterns observed by Morris and colleagues [20].

We conducted a similar set of analyses to uncover the relationship between students' age, their topic assignment from the LDA model, and the courses they took. As when analyzing gender, we did not find a 3-way interaction between these three factors. Instead, we found statistically significant relationships between all of the 2-way interactions between these factors. To study the relationship between course and topic, given age group, we used correspondence analysis. Here, we found that one course, Ochem, and a reason related to enrolling for Ochem, were far from the other courses, and the other four courses considered in this paper shared similar relationships with one another across age groups. To further understand the relationship between these three factors, we analyzed how age group and course, given their reason for taking the course, were related. When considering this relationship, we generally observed that students in the younger and older age groups enrolled in some of the courses more than expected. When more closely considering the topic from the LDA model and student age group, given the course a student took, we often found students in the younger and older age groups gave topics more or less than we would expect. This suggests that the students in this study who are in the two youngest and two oldest groups take courses and give reasons we might not expect.

Implications for course design: The finding that there is an age and gender dependence with respect to who enrolls in the courses may be interpreted as follows: Course designers could increase course effectiveness by including potentially age-relevant learning modules, such as a project or application focus for those in the degree-earning and job-seeking ages and information or lecture focus for those outside these ages. Furthermore, while the dependent relationship between reason and course suggests the obvious—learners are in different courses for different reasons—it could also be construed to mean that specific changes, such as the optional learning modules mentioned above, could improve course effectiveness.

In general, the approach in this paper can be used to characterize students' reasons for enrolling in MOOCs and subsequently to improve MOOCs. For example, students who feel isolated from their peers are often dissatisfied with their online courses. One of the potential ways of improving this situation could be to provide ways for learners who enroll to find community to connect with others who share this goal, thereby potentially ameliorating their isolation. In addition, instructional designers could help learners customize their learning experience if they knew how learners with different reasons for enrolling engaged differently with a course. For example, content choices can be categorized as being introductory and advanced, and multiple learning paths could be suggested at the outset, allowing more advanced students to jump to the appropriate content rather than have to wait or muddle through and be bored with the content that they have already mastered. As another example, those motivated to advance their job potential may be provided with assignments and projects that involve authentic work applications of the material, in contexts relevant to their particular situations. In general, understanding students' reasons for enrolling in a MOOC provides key information for improving the course and improving students' experiences with that course.

Future directions: Understanding reasons for MOOC enrollment is only one part of improving course effectiveness. Future studies in this direction should analyze how learners with different goals engage with a course in combination with their patterns of engagement while in the course, and how long they stay in the course, all towards improving learning experience for those participating in MOOCs.

6. CONCLUSION

We found that students take MOOCs for many different reasons. Although multiple-choice survey responses are useful to understand the reasons that a student might enroll in a MOOC, we found it is also feasible to use students' open-ended responses to questions that asked about why they were taking the course and what they hoped to learn. We found that some of the reasons students enrolled in these MOOCs were course specific, while others showed a general interest in learning or taking a MOOC. By examining *why* students take MOOCs, we can develop a greater understanding of what students might want when they take a MOOC. If the reasons a student takes a MOOC are more thoroughly understood, it could help explain why MOOCs have such high attrition rates and provide insight to ameliorate this issue, ultimately improving retention and learning.

7. REFERENCES

[1] A. Agresti. *Categorical data analysis.* Wiley-Interscience, 3rd. edition, 2013.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.

[3] 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, 2013.

[4] J. Chang, S. Gerrish, C. Wang, J. L. Boyd-Graber, and D. M. Blei. Reading tea leaves: How humans interpret topic models. In *Advances in Neural Information Processing Systems*, pages 288–296, 2009.

[5] 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? 2013.

[6] R. W. Crues, G. M. Henricks, M. Perry, S. Bhat, C. J. Anderson, N. Shaik, and L. Angrave. How do gender, learning goals, and forum participation predict persistence in a computer science MOOC? *ACM Transactions on Computing Education*, 2018.

[7] P. G. de Barba, G. E. Kennedy, and M. D. Ainley. The role of students' motivation and participation in predicting performance in a MOOC. *Journal of Computer Assisted Learning*, 32(3):218–231, 2016.

[8] I. Feinerer, C. Buchta, W. Geiger, J. Rauch, P. Mair, and K. Hornik. The textcat package for n-gram based text categorization in R. *Journal of Statistical Software*, 52(6):1–17, 2013.

[9] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[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*, pages 21–30. ACM, 2014.

[11] K. F. Hew and W. S. Cheung. Students' and instructors' use of massive open online courses (MOOCs): Motivations and challenges. *Educational Research Review*, 12:45–58, 2014.

[12] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[13] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1-2):177–196, 2001.

[14] K. Hornik and B. Grün. topicmodels: An r package for fitting topic models. *Journal of Statistical Software*, 40(13):1–30, 2011.

[15] J. P. Howarth, S. D'Alessandro, L. Johnson, and L. White. Learner motivation for MOOC registration and the role of MOOCs as a university 'taster'. *International Journal of Lifelong Education*, 35(1):74–85, 2016.

[16] R. F. Kizilcec and E. Schneider. Motivation as a lens to understand online learners: Toward data-driven design with the OLEI scale. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 22(2):6, 2015.

[17] S. Lê, J. Josse, F. Husson, et al. FactoMineR: An R package for multivariate analysis. *Journal of Statistical Software*, 25(1):1–18, 2008.

[18] M. Liu, J. Kang, and E. McKelroy. Examining learners' perspective of taking a MOOC: Reasons, excitement, and perception of usefulness. *Educational Media International*, 52(2):129–146, 2015.

[19] D. Meyer, K. Hornik, and I. Feinerer. Text mining infrastructure in R. *Journal of Statistical Software*, 25(5):1–54, 2008.

[20] N. P. Morris, S. Hotchkiss, and B. Swinnerton. Can demographic information predict MOOC learner outcomes. *Proceedings of the EMOOC Stakeholder Summit*, pages 199–207, 2015.

[21] M. Nikita. *ldatuning: Tuning of the Latent Dirichlet Allocation Models Parameters*, 2016.

[22] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[23] B. K. Pursel, L. Zhang, K. W. Jablokow, G. W. Choi, and D. Velegol. Understanding MOOC students: Motivations and behaviours indicative of MOOC completion. *Journal of Computer Assisted Learning*, 32(3):202–217, 2016.

[24] J. Qiu, J. Tang, T. X. Liu, J. Gong, C. Zhang, Q. Zhang, and Y. Xue. Modeling and predicting learning behavior in MOOCs. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 93–102. ACM, 2016.

[25] C. Robinson, M. Yeomans, J. Reich, C. Hulleman, and H. Gehlbach. Forecasting student achievement in MOOCs with natural language processing. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 383–387. ACM, 2016.

[26] B. Swinnerton, S. Hotchkiss, and N. P. Morris. Comments in MOOCs: Who is doing the talking and does it help? *Journal of Computer Assisted Learning*, 33(1):51–64, 2017.

[27] S. Zheng, M. B. Rosson, P. C. Shih, and J. M. Carroll. Understanding student motivation, behaviors and perceptions in MOOCs. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*, pages 1882–1895. ACM, 2015.

Understanding Student Procrastination via Mixture Models

Jihyun Park
Department of Computer Science
University of California, Irvine
Irvine, California 92697
jihyunp@ics.uci.edu

Renzhe Yu
School of Education
University of California, Irvine
Irvine, California 92697
renzhey@uci.edu

Fernando Rodriguez
School of Education
University of California, Irvine
Irvine, California 92697
fernanr1@uci.edu

Rachel Baker
School of Education
University of California, Irvine
Irvine, California 92697
rachelbb@uci.edu

Padhraic Smyth
Department of Computer Science
University of California, Irvine
Irvine, California 92697
smyth@ics.uci.edu

Mark Warschauer
School of Education
University of California, Irvine
Irvine, California 92697
markw@uci.edu

ABSTRACT

Time management is crucial to success in online courses in which students can schedule their learning on a flexible basis. Procrastination is largely viewed as a failure of time management and has been linked to poorer outcomes for students. Past research has quantified the extent of students' procrastination by defining single measures directly from raw logs of student activity. In this work, we use a probabilistic mixture model to allow different types of behavioral patterns to naturally emerge from clickstream data and analyze the resulting patterns in the context of procrastination. Moreover, we extend our analysis to include measures of student regularity–how consistent the procrastinating behaviors are–and construct a composite Time Management Score (TM). Our results show that mixture modeling is able to unveil latent types of behavior, each of which is associated with a level of procrastination and its regularity. Overall, students identified as non-procrastinators tend to perform significantly better. Within non-procrastinators, higher levels of regularity signify better performance, while this may be the opposite for procrastinators.

Keywords

Procrastination, Regularity, Time Management, Student Modeling, Clickstream Data, Online Courses, Probabilistic Mixture Model, Poisson distribution

1. INTRODUCTION

As colleges and universities continue to increase the number of online course offerings, these classes are becoming a normal part of students' learning experiences. While online courses have made learning more accessible to students, prior work suggests that students enrolled in online courses have worse learning outcomes when compared to students enrolled in face-to-face courses [2]. One important reason for this is that the online learning environment requires a higher degree of self-regulation than the face-to-face environment [5]. Students must effectively plan and regulate their learning time, and monitor their own progress in order to meet important deadlines [31], but students may lack some of these important skills. Moreover, online courses have a high degree of anonymity. Students are not physically present in a classroom, and their activity on Learning Management Systems (LMS) is not made public to the rest of the class. This absence of face-to-face accountability may cause students to disengage with the course much more than they would in traditional classrooms. The lack of structure and anonymity may lead students to procrastinate, putting off work until close to important deadlines. Therefore, understanding students' learning behaviors relating to time management, especially procrastination, could be one important mechanism for improving online learning.

Clickstream data sets have provided rich resources for analyzing students' time management behaviors. Procrastination has been measured using the specific time points at which students take certain actions within an online course, such as accessing content pages, watching lectures, and submitting quizzes. A common way to measure procrastination is to calculate the amount of time a student is engaged with the LMS prior to an important course deadline. Studies that use these types of measures as indicators of procrastination find that the indicators are negatively correlated with course outcomes [14, 16, 30]. In the context of studying planning behaviors, researchers have also developed measures of student regularity are in the timing and spacing of their course activities, and found that higher measures of regularity correlate with better performance [28, 3].

Motivated by these previous studies, we utilize clickstream data to further understand procrastination using two online classes offered at a large public university. These two classes were designed so that the students are expected to space out their studies on a daily basis, and to set weekly deadlines. In this paper, we investigate the use of probabilistic mixture modeling to analyze time-stamped logs of student activity in the context of these two online classes. The mixture model identifies different behavioral patterns in the data, where the patterns can be clearly identified as reflecting procrastinating and non-procrastinating behavior among the students. Moreover, we notice that while procrastinating students may procrastinate frequently, some may also exhibit a mix of planning and procrastinating behaviors throughout the course. To capture these nuances, we construct a composite score, which incorporates both the overall degree of procrastination and the regularity of procrastinating behaviors. This score captures behavioral differences of procrastinators, a notion which has been absent in prior research. The methodology we develop enables finer-grained analysis of procrastination and its relationship with learning outcomes, which can inform more effective instructional reforms in online learning.

The primary contributions of this work are four-fold.

- First, we develop a general data-driven method for identifying procrastination. This method analyzes counts of student activity and can work with any online course with periodic deadlines and that has corresponding time-stamped clickstream data.
- Second, we validate this method using two online university classes, and identify two distinct behavioral patterns which can be used to measure an individual student's degree of procrastination.
- Third, building off of prior measures of procrastination, we investigate the regularity of procrastinating behaviors and incorporate this information into a composite score, providing a more detailed perspective on procrastination.
- Fourth, for the two classes we analyze, we find that all of our measures of procrastination are highly correlated with course outcomes, lending support to prior theories of self-regulated learning and procrastination while also providing new insights.

2. RELATED WORK

2.1 Self-Regulation, Procrastination, and Academic Success

Self-regulated learning refers to the process of directing one's own learning experience [31] and these processes encompass several attitudes and behaviors. For instance, models of self-regulated learning generally distinguish between motivational beliefs about learning, goal setting and planning behaviors, specific learning strategies, and metacognitive monitoring processes [22]. While each of these facets play an important role in the learning process, research on online learning finds that students' planning and time management behaviors are important indicators of course success [10, 30]. Procrastination behaviors, which refer to delaying coursework until major deadlines, reflect poor planning and time management.

Several studies have focused on procrastinating as a major barrier that hinders students from succeeding in online courses [10, 29]. Using online course analytic data, one recent study found that students who did not begin working on assignments until hours before a deadline received lower course grades when compared to students who began their work earlier [9]. Other studies have found similar results, where students who delay working on assignments are more likely to perform poorly [29, 30]. These results confirm the undesirable nature of procrastination as well as the importance of regular learning behaviors.

Another extensive body of work has shown that students from underrepresented backgrounds, such as those who come from low-income households, or who are first to attend college, are a greater risk for leaving STEM majors [7]. This problem may be additionally exacerbated in online coursework. There are many important factors that explain issues surrounding underrepresented student success, such as lack of mentoring, financial concerns, and feelings of exclusion [25]. With regard to self-regulatory behaviors such as procrastination, prior work has also shown an increased tendency for underrepresented groups to engage in more procrastination than the counterparts [24]. However, this study was not conducted in an educational context and the procrastination was measured subjectively using surveys. With this in mind, a side aim of our work is to explore the relationship between individual differences in procrastination (time management behavior, in general) and students' external background characteristics, specifically for the students taking online courses.

2.2 Measuring Procrastination and Regularity

Measures of procrastination are relatively straightforward and similar across various learning environments. In the most common measures, researchers capture the time that students finish a certain task and calculate the difference between this time and either the release time [3] or the deadline [16, 14] of the task. This type of measure has the merit of being very interpretable, but a limitation is that it only captures the average degree of procrastination without depicting nuanced patterns in these behaviors.

Regularity, on the other hand, is a higher-order concept that allows for different definitions. Accordingly, there has been a slightly larger pool of measures in the literature. Some studies define regular behaviors as repeating certain temporal patterns in a cyclic manner, and apply methods from signal processing to model hidden frequencies within students' behavioral streams [27, 3]. Another popular way of operationalizing regularity is to relate regularity to changes of learner behaviors, and quantify the changes via measures of variation [1, 28, 23] or explicit statistical modeling [21]. These different definitions are not exclusive and share many similar properties.

Most of the existing studies regarding time management in online learning examine either procrastination or regularity, and those that investigate both treat them as independent features of student behaviors. Our work extend these studies by understanding how regularity and procrastination are interrelated.

2.3 Cluster Analysis and Mixture Modeling

Clustering in general is a widely used technique in data analysis for automated data-driven discovery of groups or clusters in data. In the context of analyzing education data, clustering algorithms have found broad application as a technique for clustering of students into groups based on their behavioral patterns. For example, Toth et al. [26] cluster students based on their problem-solving interaction patterns using the X-means algorithm (a variation of the well-known K-means clustering algorithm) for a better understanding of complex problem solving behaviors and identifying levels of problem solving proficiency. Ng, Liu, and Wang [20] use survey scores of motivated strategies for learning questionnaires to cluster students into multiple groups. The resulting groups obtained by hierarchical clustering with Ward's method exhibit distinct learning profiles of motivational beliefs and self-regulatory strategies.

The clustering approach we follow in this paper is probabilistic model-based clustering [12, 19]. In this framework, each cluster corresponds to a probability distribution (also known as a "component") in a mixture model and the entities being clustered are assumed to have been generated by one of the component distributions. This probabilistic framework for clustering has a number of advantages over non-probabilistic techniques such as K-means clustering or hierarchical clustering. For example, as we describe later in the paper, the framework allows us to model count data in a natural manner using Poisson distributions as components in the mixture model, where each component (or cluster) represents a different Poisson distribution over count outcomes. The Poisson mixture model has been applied to a number of different fields including marketing [4], finance [15], biology and bioinformatics [6, 11], document analysis [17], and so on. However, to our knowledge, there has not been any prior work on the development of Poisson mixtures in an education context, particularly for the problem of clustering students based on their observed activity in online classes.

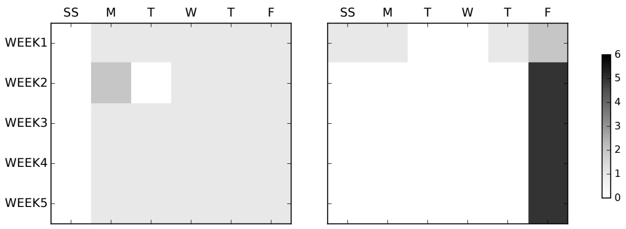


Figure 1: Examples of student ***daily activity counts*** **(specifically, the number of video watching tasks per day) displayed as a matrix of week × day counts. SS indicates Saturday and Sunday.**

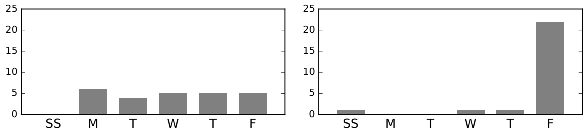


Figure 2: ***Aggregated daily task counts*** **across weeks ($\boldsymbol{y_i}$) for the two students shown in Figure 1.**

3. METHODS

3.1 Student Activity Counts

For courses where time-stamped student-generated events are tracked via logs of clickstream data, we can count these events on a daily basis. Thus, we can get a set of *daily activity counts* for each student throughout a course, where the activity can correspond to specific types of tasks of interest (such as video-watching, quiz submission, and so on) Figure 1 shows *daily activity count* data for 2 students from one of our course data sets. The data for each student is displayed as a matrix, where the grayscale indicates the number of tasks performed by each student on each day over the 5 week duration of the course. This type of display is useful in terms of capturing the temporal aspect of when a student is engaged in a particular activity such as watching a lecture video or submitting a quiz. It also indicates that one of the students (on the right) may be procrastinating each week—we discuss these types of patterns in more detail later in the paper.

We can also compute the sum over weeks to get the *aggregated daily counts* assuming that there is a structure in the course that repeats every week (which is the case for the two courses we study in this paper). Examples of the *aggregated daily counts* are shown in Figure 2 as bar plots, computed by aggregating across the weekly rows of data for each student in Figure 1.

3.2 Mixture Model with Gamma Priors

In this section we discuss our use of a Poisson mixture model to cluster students based on their activity counts, focusing on the *aggregated daily counts* as in Figure 1. In terms of notation we let $\boldsymbol{y_i}$ be the vector of *aggregated daily counts* for student i, where $i = 1, \ldots, N$. The dimensionality D of each vector is the number of days ($D = 6$ in this case since Saturday and Sunday are collapsed into one). Thus, our data consists of N students each with a D-dimensional vector of *aggregated daily counts.*

To model this data we use a probabilistic mixture model with Poisson components. Let K be the number of components (or clusters) with an index $k = 1, 2, \ldots, K$. The unobserved latent variable z_i takes values from the set $\{1, \ldots, K\}$ and corresponds to the latent component or cluster that student i is presumed to belong to. Each of the k components consists of a vector of Poisson rate parameters, $\boldsymbol{\lambda_k} = [\lambda_{k1}, \ldots, \lambda_{kd}, \ldots, \lambda_{k6}]$, where d from $\{1, \ldots, 6\}$ represents a specific day of the week. For example, one component could have very low values for all the λ_{kd}'s, for students with low daily activity, and another component could have high values for all the λ_{kd}'s, for students with high daily activity.

When fitting our mixture model to data, we take a Bayesian approach [13] and use Gamma prior distributions for the rate parameters λ_{kd}. The primary reason for doing this is to encourage the model to avoid degenerate solutions with a small component that has one or more rate parameters λ_{kd} at or near a value of 0. This can produce a high-likelihood solution but one that is not useful. In our experimental results later in the paper we used hyperparameters of $\alpha = 1.1$ and $\beta = 0.1$ for the Gamma distribution. These hyperparameter choices have the effect of making the Gamma prior behave like a step function, putting zero probability mass at $\lambda_{kd} = 0, k = 1, \ldots, K, d = 1, \ldots, 6$, and a relatively flat uninformative prior distribution over positive rate parameter values. Figure 3 depicts a graphical model representation of the Poisson mixture model with a Gamma prior on the $\boldsymbol{\lambda}$ parameters for each component.

The likelihood for the data $\boldsymbol{y_i}$ for each student i under this mixture model can be written as

$$p(\boldsymbol{y_i}|\boldsymbol{\lambda}) = \sum_{k=1}^{K} p(\boldsymbol{y_i}|z_i = k, \boldsymbol{\lambda_k})p(z_i = k) \qquad (1)$$

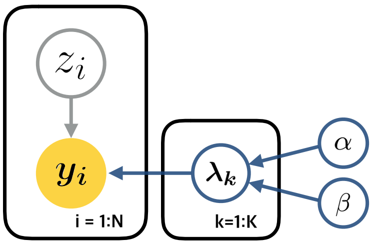


Figure 3: Graphical representation of the Poisson mixture model with Gamma prior. $\boldsymbol{y_i}$ and $\boldsymbol{\lambda_k}$ are 6 dimensional vectors. N is the number of students, and K is the number of mixture components.

where $p(z_i = k)$ is the marginal mixing weight for each component, and each component distribution can be written as

$$p(\boldsymbol{y_i}|z_i = k, \boldsymbol{\lambda_k}) = \prod_{d=1}^{D} p(y_{kd}|\lambda_{kd}, z_i = k) \tag{2}$$

assuming conditional independence of the daily counts y_{kd} given component k. λ_{kd} is the rate for day d for component k and each distribution $p(y_{kd}|\lambda_{kd}, z_i = k)$ is a Poisson distribution. The prior distribution is defined as a product over independent Gamma priors, one for each λ_{kd}, each with parameters $\alpha = 1.1$ and $\beta = 0.1$.

3.3 Learning Parameters with the EM Algorithm

To estimate the parameters $\boldsymbol{\lambda_k}$ of our model we use the Expectation-Maximization (EM) algorithm, an iterative algorithm that is widely used in fitting mixture models to data [8, 18]. More specifically, we use the EM algorithm to maximize the product of the data likelihood times the prior (both defined above). This results in both (a) maximum a posteriori (MAP) parameter estimates for the Poisson components in the model, and (b) membership weights w_{ik} that reflect the probability (under the fitted model) that each student i belongs to component (or cluster) k.

Each iteration of the EM algorithm consists of two steps, the E (expectation) step and the M (maximization) step. In the E-step, conditioned on some fixed (current) values of the parameters, the probability of membership w_{ik} is computed for each component $k = 1, \dots, K$, for each student $i = 1, \dots, N$.

$$\begin{aligned} w_{ik} &= p(z_i = k|\boldsymbol{y_i}, \boldsymbol{\lambda}, \alpha, \beta) \\ &\propto p(\boldsymbol{y_i}, z_i = k, \boldsymbol{\lambda_k}|\alpha, \beta) \\ &\propto p(\boldsymbol{y_i}|z_i = k, \boldsymbol{\lambda_k})p(\boldsymbol{\lambda_k}|\alpha, \beta)p(z_i = k) \end{aligned} \tag{3}$$

These membership weights are important in our later analyses, since they provide information of how likely it is that each data point i (in our case, student i) was generated by component k. In the M-step, conditioned on the set of membership probabilities w_{ik}, a point estimate of each parameter is estimated via MAP estimation.

$$\hat{\boldsymbol{\lambda}}_{\boldsymbol{k}} = \frac{\sum_i w_{ik}(\boldsymbol{y_i} + \alpha - 1)}{\sum_i w_{ik}(1 + \beta)} \tag{4}$$

$$\hat{p}(z_i = k) = \frac{\sum_i^N w_{ik}}{N} \tag{5}$$

These MAP parameter estimates provide the input for the next E-step, and thus, the cycle of E and M-steps continue iteratively.

The EM algorithm as a whole consists of randomly initializing the parameters of the model, followed by repeated computation of pairs of E and M steps, until the log-likelihood is judged to have converged (i.e., when the improvement in log-likelihood from one iteration to the next is less than some small value ϵ, or when the average membership probability value is not changing significantly from one iteration to the next).

Python code for this EM algorithm is available online at `https://github.com/jihyunp/student_poisson_mixture`.

4. DATA SETS

Two data sets from the same undergraduate online course were used in this study: one from summer 2016, and the other from summer 2017. Both summer courses were 5 weeks long. While each class was taught by two different instructors, the class content, such as the lecture videos, resources, and assignments, were the same. In both classes, students were assigned 5 video lectures every week and each lecture video had a corresponding quiz. The instructors encouraged students to watch one lecture video and complete the corresponding quiz each day, from Monday through Friday.

Although students were encouraged to follow this schedule, the actual deadline for watching the 5 lecture videos and completing the quizzes was on Fridays at midnight. While this structure gave students freedom to watch the lecture videos when they wanted, this flexibility also allowed them to procrastinate.

Most of the students' activities were recorded through the Canvas Learning Management System (LMS). These activities included downloading course content, watching lecture videos, taking online quizzes, submitting assignments, etc. Every time a student clicked on a URL within the Canvas system, the click event was logged with the student ID, URL, and time-stamp. The clickstream data was processed so that it only focused on the activities of daily tasks, resulting in *daily activity counts*, as mentioned in the previous section (Figure 1 and Figure 2). Only one event per task was counted and thus the sum of the matrix for each student was 25 or less (for 5 video lectures × 5 weeks). We chose to count only the first attempt (first click event) for each task.

In addition to the clickstream data, student demographic data was available through the university's institutional research office. It included both demographic information (gender, ethnicity, first generation status, low income status, and full-time status) and prior academic achievement (total SAT[1] score). Some students did not agree to provide this demographic data, although most did. For this reason, our later analyses based on demographic information are based on the subset of students who agreed to share this information.

[1] A standardized test widely used for college admissions in the United States.

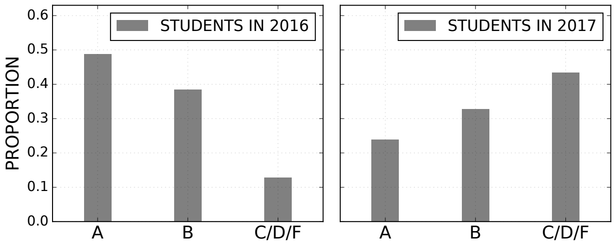


Figure 4: Grade distributions of students in 2016 class (left) and in 2017 class (right). Two classes show very different grade distributions. Almost half of the students received an A for the class in 2016, whereas more students got lower grades in 2017.

Although both classes used the same materials and implemented the same deadlines, there were some notable differences in how the the click events were recorded, as well as how each instructor structured the course. We describe these differences in the following sections.

4.1 Class in 2016

Online lectures and daily quizzes were offered outside the Canvas LMS for this class. Each lecture video was embedded on a separate web page on the server that we had access to, and the links to the web pages were provided via the Canvas weekly module.

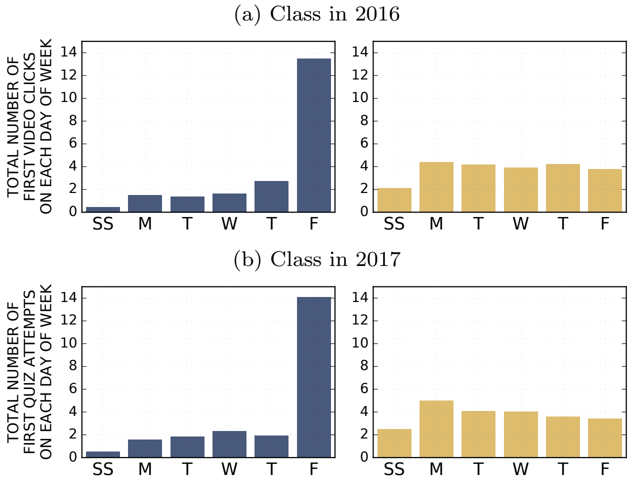
Logs for the daily quiz attempts were not accessible, so instead we used the first "video clicks," which are from the logs of HTTP GET requests of the video embedded web pages. For each student, we matched the IP addresses of the video logs (from the server) with the IPs recorded on the Canvas LMS.

After removing 4 students with very low activity (0 or 1 video clicks in total) there were 172 students with activity counts available for analysis. More than 90% of the students received a passing grade, and half of the students received an A (Figure 4). Completing the daily tasks (watching videos and solving quizzes) counted as 15% towards the overall grade for each student.

4.2 Class in 2017

The video click logs for this class were not available since the videos were uploaded on a third-party server. However, the daily quizzes that students took after watching the lecture videos were recorded through the Canvas system, and we were able to obtain students' quiz submission time-stamps via the corresponding clickstream data. Therefore, for this class we focused on the first clicks for daily "quizzes." Note that this is different from the 2016 class data, which used the first clicks for each video-watching event.

There were 145 students in the class—we used data for 140 students after dropping 5 students with very low activity (as with the 2016 class). As previously noted, a different instructor taught the class in 2017 than in 2016. The instructor for the 2017 class changed the contribution to 8% of the total grade for watching and completing the lecture videos, significantly less than in the 2016 class (15%). The grade distribution of the 2017 class in Figure 4 is also significantly different to that in 2016—there are significantly fewer students who received A's or B's in 2017 compared to the 2016 class.


(a) Class in 2016

(b) Class in 2017

Figure 5: Poisson mixture component means ($\boldsymbol{\lambda}_k$'s) from modeling aggregated daily task counts ($\boldsymbol{y}_i$) for the class in 2016 (upper) and 2017 (lower).

5. PROCRASTINATION AS A MIXTURE COMPONENT

Below, we present and discuss the results of fitting a two-component ($K = 2$) Poisson mixture model to the *aggregated daily task counts* for the two classes described in section 4. We also explored models with more components, $K = 3, 4, \ldots$, but found that the $K = 2$ model broadly captured the primary modes of student behavior and that higher values of K tended to split the two main modes into further subgroups without providing any significant additional insight.

Figure 5 shows the expected number of counts per group, i.e., the rate parameters, $\boldsymbol{\lambda}_k$'s. The two group-dependent rate patterns across the days of the week, for both 2016 and 2017, show two very distinct behavioral patterns. One of the mixture components has a very high rate on Friday and low rates on the other days of the week. The other component has low and relatively flat rates from Monday to Friday. Considering the fact that the deadline for daily tasks in these courses is on Fridays, these two patterns clearly reflect two different types of student behaviors: *procrastination* and *non-procrastination.*

5.1 Characteristics of the Two Behavioral Groups

We can threshold the membership weights at 0.5 to classify each student $i = 1, \ldots, N$ into one of the two groups, i.e., if $w_{i1} > 0.5$ then student i is assigned to the *procrastination* group (where $k = 1$ corresponds to the *procrastination* group). About 36-37% of the students were assigned to the *procrastination* group in each of the two years.

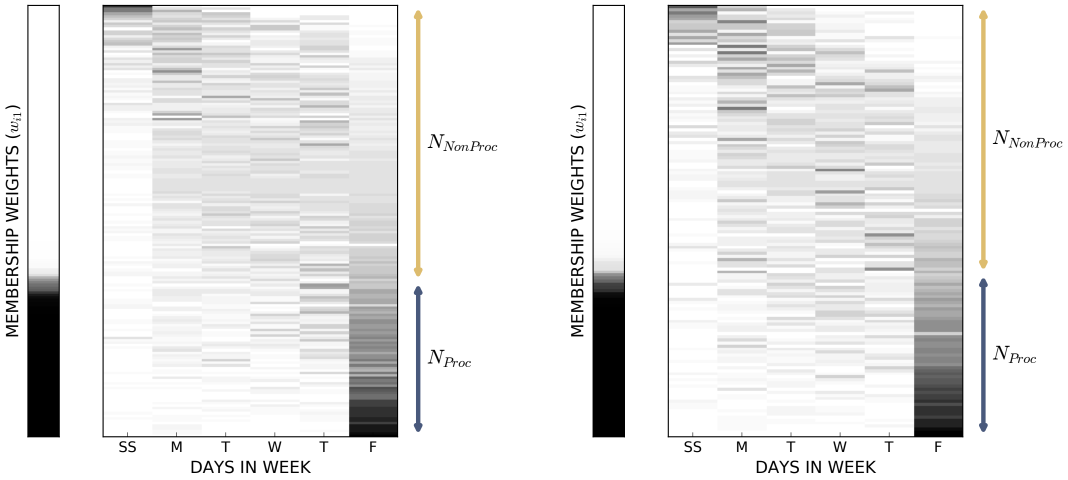


Figure 6: Aggregated daily task counts shown along with the membership weights. Each row represents a student, and the students are sorted by the membership weight w_{i1}. The left figure is for the class in 2016, and the right figure is for the class in 2017.

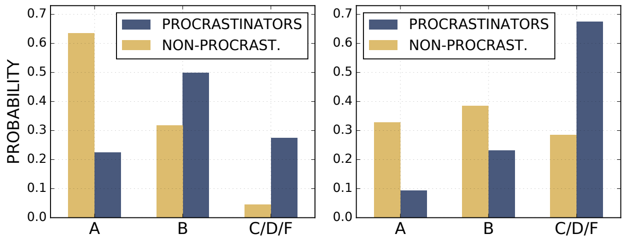


Figure 7: Probability of receiving each grade given that the student is in the *procrastination* group or in the *non-procrastination* group in 2016 (left) and in 2017 (right).

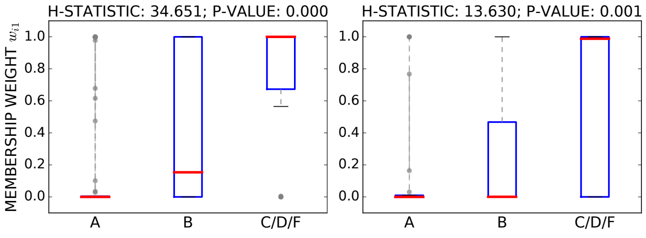


Figure 8: Distribution of w_{i1} in different grade groups of class in 2016 (left) and in 2017 (right). H-statistic comes from a Kruskal-Wallis test. (w_{i1}: membership weight on the procrastinating group)

The two plots in Figure 6 illustrate the students' week-aggregated activities along with the students' membership weights. Each row in each plot represents a student and the wider matrix plot shows the aggregated daily counts, sorted by their membership weight w_{i1}. The values in the matrix range from 0 to 25 and a darker color means that there are more task activities on that day of the week. The two plots from different years look almost identical and they clearly show the two types of behavior. The students (rows) at the bottom of each plot have more counts (darker colors) on Fridays and belong to the *procrastination* group. There is also a small group of students at the top of both plots who tend to be more active over the weekend. The size of this group of students is relatively small and their behavior pattern is effectively that of *non-procrastinators* since they are the "early birds" who check out the lecture videos or the quizzes early in the week.

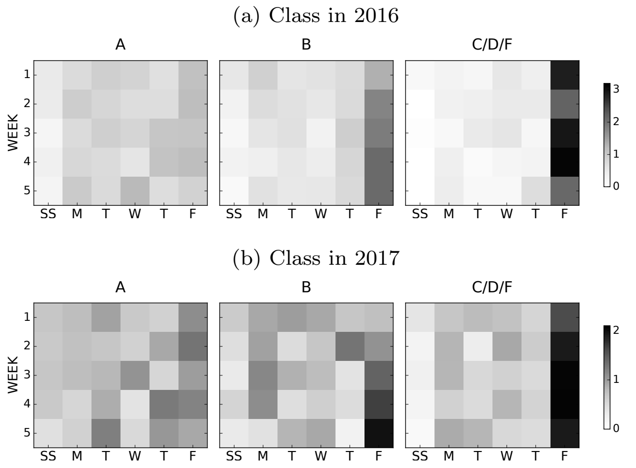
The membership weights are shown on the narrower bar plot (left of each year's plot), where a darker color represents a higher membership weight of belonging to the procrastination group (with a weight close to 1). We can observe that there is a relatively small amount of grey area in the bar plot (for both years), which means that the majority of the students have a very high probability of being assigned to one group or the other.

5.2 Association between Behaviors and Grades

We can further analyze the relationship between the two different behavioral groups and the grades. We show the grade distribution in each group in Figure 7. Results from the two classes are shown side by side. It is obvious from the figure that the *non-procrastinators* tend to get significantly more A grades than the *procrastinators*, whereas the *procrastinators* get more C, D, and F's. Even though the overall grade distributions were quite different in the two classes (see Figure 4), we find a strong correlation between the behavioral groups and course outcomes. In both classes, the non-procrastinating students are about three times more likely to get an A grade than the procrastinating students. These probabilities were significant at the 0.01 level using a chi-squared test.

We can further analyze the relationship between procrastination behavior and grade outcomes by grouping students by their grade (rather than by the behavioral group) and looking at the patterns of behavior for each grade group.

As we saw in Figure 4, a majority of the students got a passing grade in 2016. The number of students who received A,


(a) Class in 2016

(b) Class in 2017

Figure 9: The number of task counts per day, for each of the 5 weeks, averaged over the students in each grade group. Left: students who received A's, middle: students who received B's, right: students who received a C, D, or F.

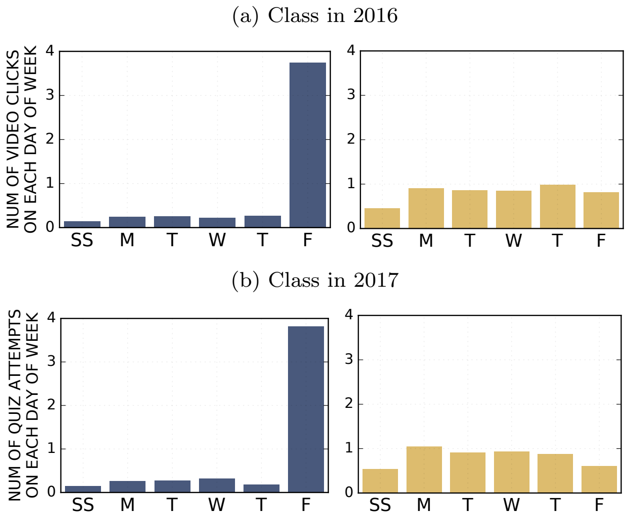
B, and the others (C, D, or F) were 84, 66, and 22, respectively. The left boxplot in Figure 8 informs that "A students" have very low membership weight (w_{i1}) values, but the "C, D, F students" have very high membership weight values. This can be interpreted as saying that the students who received lower grades (C, D, or F) have higher probabilities of being *procrastinators*.

We can see the similar result for the class in 2017 from the right side of the plot in Figure 8. There were 27, 37, 49 students in each of the grade groups (there were 27 students whose grade information was unavailable). The broader distribution of weights in the C, D, F group may be due to the fact that there were many more students with lower grades than higher grades in this year of the course.

The association between behavior and grade outcome is also clearly visible in the raw data, i.e., the task activity counts, for both years. Figure 9 clearly illustrates the behavior patterns for students with different grades. We can see a very dark color on Fridays on the matrices on the right side (students who received C, D, or F grades), and more evenly distributed colors on the left matrix plots, which shows the activities of students who received A grades.

6. REGULARITY OF PROCRASTINATION

In the previous section we showed that Poisson mixture modeling can help to unveil two latent types of students: *procrastinators* and *non-procrastinators*. Because these results are based on modeling *aggregated* daily activity counts across multiple weeks, they do not shed light on how students might change their procrastinating behavior over time during different stages of a course. For example, a student who is generally a procrastinator might only procrastinate every other week, while a non-procrastinator might postpone studying during some week. To gain insights into these nuances, we investigate the *regularity of procrastination* in this section.


(a) Class in 2016

(b) Class in 2017

Figure 10: Poisson mixture component means (λ_k's) from modeling individual week of daily task counts (y_{ij}) for the class in 2016 (upper) and 2017 (lower). The number of first clicks on any lecture video in 2016, and the number of first attempts on any quiz in 2017, are modeled.

6.1 Regularity across Weeks

We focus here on inter-week regularity, which is defined as the extent that students repeat their behavior across different weeks. We use the same Poisson mixture modeling methodology described earlier in the paper except that we model each *individual week* of daily activity counts for each student rather than aggregating across weeks. The resulting mixture components are similar to the aggregated case in that there are two distinct weekly behaviors, *procrastination* and *non-procrastination* (see Figure 10). Each week of a student's behavior is modeled as being generated by one of the two components in the model, and we can estimate the membership weight of belonging to the *procrastination* group (or component) for each week for each student, i.e., w_{ij1} for student $i = 1, \ldots, N$ and week $j = 1, \ldots, M$ where $M = 5$ is the number of weeks.[2]

To quantify student i's regularity, we use the standard deviation of the *procrastination* weights across weeks:

$$SD_i = \left(\frac{1}{M-1} \sum_{j=1}^{M} (w_{ij1} - \overline{w}_{i\cdot 1})^2 \right)^{1/2} \tag{6}$$

where w_{ij1} and $\overline{w}_{i\cdot 1}$ represent the student-week membership weights for the *procrastination* component in week j, and the average of those weights across the M weeks, respectively.

[2] We could also use the non-aggregated *weekly* daily activity counts for the earlier group analyses in Section 5. Instead of the membership weights w_{i1} for student i, the mean value of the M membership weights ($\overline{w}_{i\cdot 1}$) could be used for thresholding. This would allow us to use the same analyses in Section 5 and 6 by fitting a single mixture model. We investigated this and found the results were almost identical to those reported in the paper. Given this, for the investigation of regularity we used weekly activity counts to see changes in weekly behavior, and for overall clustering (Section 5) we used total aggregate counts for ease of interpretation.

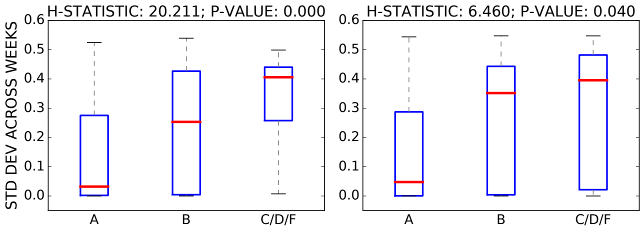


Figure 11: Distribution of SD_i in different grade groups of class in 2016 (left) and in 2017 (right). H-statistic comes from a Kruskal-Wallis test. (SD_i: inter-week standard deviation of w_{ij1}, membership weight on the *procrastination* group in week j)

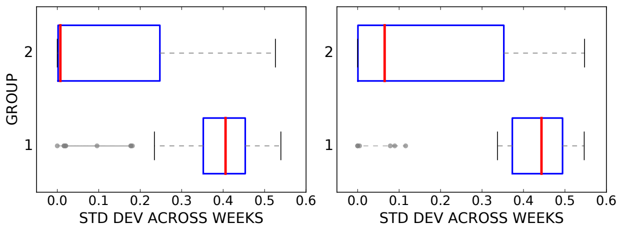


Figure 12: Distribution of SD_i in two behavioral groups (see Figure 6) of class in 2016 (left) and in 2017 (right). Within each subgraph, the *procrastination* group is indexed as 1, while the *non-procrastination* group is indexed as 2. (SD_i: inter-week standard deviation of w_{ij1}, membership weight on procrastination group in week j)

By definition, a higher value for SD_i signifies more volatile behavioral patterns.

In light of prior research, regularity is strongly correlated to performance [3]. We plot the distribution of SD_i's within three grade groups in Figure 11. Consistent with prior findings, students with better grades in general have lower levels of SD_i, hence are more regular learners. More formally, we perform a Kruskal-Wallis test within each class, with results reported above the graph. In both years, the three groups have significantly different SD_i distributions.

6.2 Incorporating Regularity and Procrastination

In previous sections, w_{i1} and SD_i capture different dimensions of procrastinating behavior, and their interaction is worth discussing further. For one thing, procrastinators and non-procrastinators may have different levels of regularity. We compare the distribution of SD_i within each behavioral group (assigned identically as in Figure 6) and plot the results in Figure 12. Common to both classes, *procrastinators* are centered around 0.4, while *non-procrastinators* on average have very small values below 0.1. We also calculate Pearson's correlation coefficient between w_{i1} (continuous membership weights before hard group assignments, as defined in Section 5) and SD_i, resulting in values of 0.675 for 2016 and 0.590 for 2017, both statistically significant at the 0.001 level. From these results, we can conclude that non-procrastinating students are also more likely to stay consistent throughout the course, while procrastinators jump between spacing out their studies and postponing everything until the last day. On the other hand, the relationship between regularity and academic performance may substantially vary depending on how much a student is a procrastinator. Procrastinators who put off studying as a habit (with high regularity) may be more at-risk than those who occasionally jump to a spaced-out pattern, while this is the opposite for non-procrastinating students. To incorporate this asymmetry, we attempt to define a single index built upon w_{i1} and SD_i. As these two measures are both conceptually related to time management abilities, we name the index to be the Time Management Score (TM_i). To reflect their interaction, we multiply variations of w_{i1} and SD_i for each student i. Since w_{i1} and SD_i are both negatively correlated with outcome, we use the negative of their values in the index to allow for more natural interpretation. Moreover, because the score should be weighted in oppo-

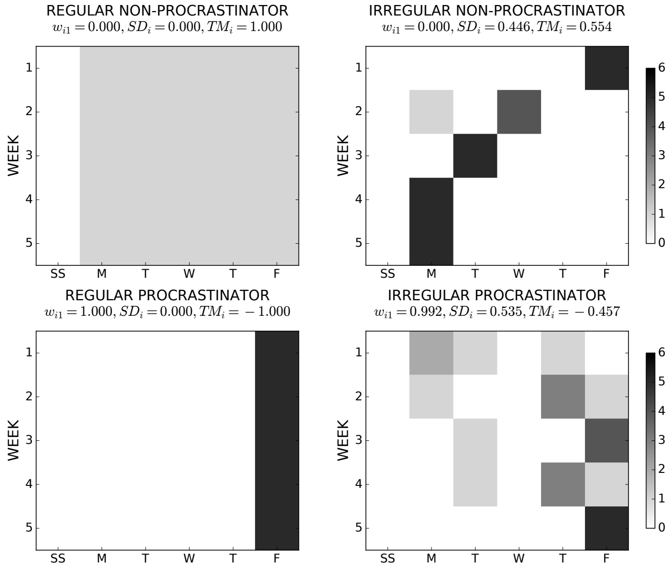


Figure 13: Number of daily activity counts for four prototypical students in the 2016 class. (w_{i1}: aggregated membership weight on *procrastination* group; SD_i: inter-week standard deviation of w_{ij1}, membership weight on *procrastination* group in week j; TM_i: Time Management Score)

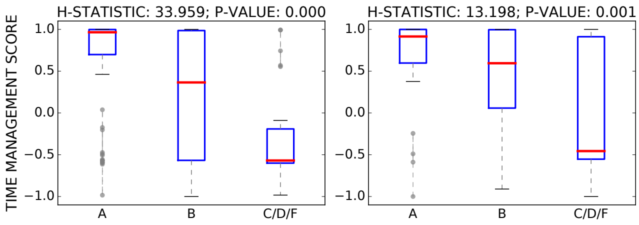


Figure 14: Distribution of Time Management Score (TM_i) in different grade groups of class in 2016 (left) and in 2017 (right). H-statistic comes from a Kruskal-Wallis test.

site directions depending on the student's behavioral group, we made variations to w_{i1} so that the most procrastinating student with the same degree of regularity would have the smallest score. Taking all of the above into account, we define TM_i as follows:

$$\begin{aligned} TM_i &= (1 - w_{i1})(1 - SD_i) + [-w_{i1}(1 - SD_i)] \\ &= (1 - 2w_{i1})(1 - SD_i) \end{aligned} \tag{7}$$

To evaluate the validity of this index, we examine whether its properties are aligned with theoretical assumptions. As discussed above, from the perspective of academic success, higher regularity is a negative behavioral feature for procrastinators but is a positive feature for non-procrastinators. In this context, it is natural to investigate how regularity and procrastination affects the value of TM_i, and how this value relates to desirable and undesirable outcomes.

From Equation (7), we know that the value of TM_i is positive or negative depending on whether w_{i1} is greater than 0.5 or not. Thus, $w_{i1} = 0.5$ is the watershed of whether SD_i positively or negatively contributes to TM_i. Given that our threshold for hard group assignment in Section 5 is also 0.5, the interpretation is straightforward: higher regularity leads to higher TM_i within the *procrastination* group, and it is more so for "purer" procrastinators; the opposite story can be told within the *non-procrastination* group.

For an intuitive examination, we choose four prototypical students with different levels of procrastination and regularity from the classs in 2016, and plot the daily counts of their first video clicks in Figure 13, along with their w_{i1}, SD_i and TM_i. As we would expect, non-procrastination with high regularity (upper-left), the most desirable pattern, has $TM_i = 1$, the maximum value possible in our context. By contrast, the regular procrastinator (lower-left) gets the minimum value of $TM_i = -1$. The remaining two students with similarly low regularity have TM_i values between the two extremes, but are respectively closer to the one that belongs to the same behavioral group. In a word, these visual patterns further validate the construction of TM_i, which more precisely measures the degree of procrastination by incorporating regularity information.

To determine if TM_i captures the desirability of certain procrastinating patterns, we probe into the relationship between this index and course outcomes. Similar to what we did earlier with w_{i1} and SD_i individually, we plot the distribution of TM_i within three grade groups. As shown in Figure 14, there exists a positive relationship between TM_i and performance, which is statistically significant under a Kruskal-Wallis test. The TM_i score incorporates two measures (w_{i1} and SD_i) and amplifies the information that is potentially predictive of performance, providing a more nuanced view of procrastination.

7. RELATIONSHIP WITH STUDENT BACKGROUND

Having explored the fine-grained differences in students' procrastinating behaviors and their relationship with outcomes, we want to further examine if these variations can be discriminated by students' background characteristics. The goal of this analysis is to understand whether there exists a potential risk factor among underrepresented students, or if instead, the behavioral differences we observe are more individual-level in nature. We also sought to explore whether prior academic achievement could explain differences in procrastinating behaviors.

Table 1: Relationship between demographic variables and procrastination/regularity measures for the 2016 class

(a) Behavioral group assignment (binary)

Demographics	N	Test	p-value
FirstGen	144	χ^2-test	0.566
LowInc	151		0.672
SAT	147	K-W test	0.238

(b) SD and TM (continuous)

Demographics	N	Test	SD p-val	TM p-val
FirstGen	144	K-W test	0.884	0.954
LowInc	151		0.175	0.294
SAT	147	Pearson's r	0.118	0.363

Table 2: Relationship between demographic variables and procrastination/regularity measures for the 2017 class

(a) Behavioral group assignment (binary)

Demographics	N	Test	p-value
FirstGen	120	χ^2-test	0.218
LowInc	128		0.955
SAT	125	K-W test	0.802

(b) SD and TM (continuous)

Demographics	N	Test	SD p-val	TM p-val
FirstGen	120	K-W test	0.136	0.897
LowInc	128		0.754	0.973
SAT	125	Pearson's r	0.505	0.820

From a comprehensive list of demographic variables, we choose three that are of general interest in education research: *Low Income Status*, *First Generation* and *Total SAT Score*. The first two binary variables represent a student's social-economic status, and the last continuous variable is a proxy for prior academic achievement.

We separately test the relationships between these three variables and three measures of procrastination and regularity in previous sections: behavioral group assignment (as in Section 5.1), SD and TM (as in Section 6). The specific statistical tests we use and their results are reported in Table 1 for the class in 2016, and Table 2 for the class in 2017. Because the demographic information contains missing values, we only include students who have relevant information in each of the tests (the number of students, N, is reported in the tables).

The results show that for both classes none of these demographic variables have any significant relationship with procrastination and/or regularity. This suggests that failures in time management may arise more from students' inherent factors than specific background characteristics, and that effective instructional interventions are less likely to be hampered by students' underrepresented backgrounds. However, due to the limited class sizes, this inference still needs to be further explored at scale.

8. CONCLUSIONS

In this paper, we introduce a data-driven methodology for characterizing student procrastination in online courses. Based on Poisson mixture modeling, the proposed approach can be applied to courses where tasks with clear deadlines are regularly assigned and students' timestamped activites related to those tasks are recorded. In our experiments with two undergraduate online classes, this method identifies two distinct patterns in students' weekly planning behavior, which can be further utilized to measure procrastination. This measure is found to be strongly correlated with course outcomes for both classes. In addition, our proposed Time Management Score (TM) is able to quantify students' overall time management skills by combining overall degree of procrastination with the regularity of the behavior. Interestingly, while TM is a strong predictor of course outcomes, it is not significantly related to students' demographics or prior academic achievement. These results suggest that, as a whole, procrastination behaviors seem to be more of an inherent characteristic.

These types of clickstream data and analyses allow for rich complements to other types of educational research. For example, the proposed behavioral measures of time management can be combined with survey data to examine how accurate students' perceptions of their skills are, and to identify students who might be especially prone to benefit from support. From the practical perspective, these data-driven approaches can be incorporated into learning management systems and work in real time. This would potentially facilitate automated assessment and intervention regarding time management skills.

There are also a number of potentially useful extensions to the methodological approach proposed here. For example, the mixture components in the two classes that we analyzed are straightforward to interpret with regard to procrastination, but this might not be the case for different course designs and structures. In these broader scenarios, it may be useful to incorporate informative Gamma prior distributions into the mixture model, with, for instance, three prior components for procrastination behavior, non-procrastination behavior, and mixed behavior respectively.

9. ACKNOWLEDGMENTS

This paper is based upon work supported by the National Science Foundation under Grants Number 1535300 (for all authors) and 1320527 (for PS). The authors would like to thank the Durendal Van Huynh, Kelsey Hollis Layos, and Qiujie Li for their assistance in acquiring the data used in this paper and giving helpful advice. We would also like to thank the course instructors, Dr. Jeremy Eaton and Dr. Anna Kwa, for their support and intellectual assistance.

10. REFERENCES

[1] A. R. Anaya and J. G. Boticario. A data mining approach to reveal representative collaboration indicators in open collaboration crameworks. In *Proceedings of the 2nd International Conference on Educational Data Mining*, pages 210–219, Cordoba, Spain, 2009.

[2] E. P. Bettinger, L. Fox, S. Loeb, and E. S. Taylor. Virtual classrooms: How online college courses affect student success. *American Economic Review*, 107(9):2855–75, 2017.

[3] M. S. Boroujeni, K. Sharma, Ł. Kidziński, L. Lucignano, and P. Dillenbourg. How to quantify student's regularity? In *Proceedings of the 11th European Conference on Technology Enhanced Learning (EC-TEL 2016)*, pages 277–291, Lyon, France, sep 2016. Springer, Cham.

[4] T. Brijs, D. Karlis, G. Swinnen, K. Vanhoof, G. Wets, and P. Manchanda. A multivariate Poisson mixture model for marketing applications. *Statistica Neerlandica*, 58(3):322–348, 2004.

[5] J. Broadbent and W. Poon. Self-regulated learning strategies & academic achievement in online higher education learning environments: A systematic review. *The Internet and Higher Education*, 27:1–13, 2015.

[6] J. M. Calabrese, J. L. Brunner, and R. S. Ostfeld. Partitioning the aggregation of parasites on hosts into intrinsic and extrinsic components via an extended Poisson-gamma mixture model. *PLoS ONE*, 6(12):1–9, 2011.

[7] X. Chen and C. Carroll. Fgs in post-secondary education: A look at their college transcripts (nces 2005-171). us department of education. *National Center for Education Statistics. Washington, DC: US Government Printing Office. Retrieved from http://nces. ed. gov/pubs2005/2005171. pdf*, 2005.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38, 1977.

[9] T. Dvorak and M. Jia. Online work habits and academic performance. *Journal of Learning Analytics*, 3(3):318–330, 2016.

[10] G. C. Elvers, D. J. Polzella, and K. Graetz. Procrastination in online courses: Performance and attitudinal differences. *Teaching of Psychology*, 30(2):159–162, 2003.

[11] W. Feng, Y. Liu, J. Wu, K. P. Nephew, T. H. M. Huang, and L. Li. A Poisson mixture model to identify changes in RNA polymerase II binding quantity using high-throughput sequencing technology. *BMC Genomics*, 9(Suppl 2):S23, 2008.

[12] C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.

[13] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian data analysis.* CRC press, 2013.

[14] S. L. Hotle. *Applications of clickstream information in estimating online user behavior.* PhD thesis, Georgia Institute of Technology, 2015.

[15] R. R. Jayasekare, R. Gill, and K. Lee. Modeling discrete stock price changes using a mixture of Poisson distributions. *Journal of the Korean Statistical Society*, 45(3):409–421, 2016.

[16] A. M. Kazerouni, S. H. Edwards, and C. A. Shaffer. Quantifying incremental development practices and their relationship to procrastination. In *Proceedings of the 2017 ACM Conference on International Computing Education Research*, ICER '17, pages 191–199. ACM, 2017.

[17] J. Li and H. Zha. Two-way Poisson mixture models for simultaneous document classification and word clustering. *Computational Statistics & Data Analysis*, 50(1):163–180, 2006.

[18] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*, volume 382. John Wiley & Sons, 2007.

[19] P. D. McNicholas. Model-based clustering. *Journal of Classification*, 33(3):331–373, 2016.

[20] B. L. L. Ng, W. C. Liu, and J. C. K. Wang. Student motivation and learning in mathematics and science: A cluster analysis. *International Journal of Science and Mathematics Education*, 14(7):1359–1376, 2016.

[21] J. Park, K. Denaro, F. Rodriguez, P. Smyth, and M. Warschauer. Detecting changes in student behavior from clickstream data. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference on - LAK '17*, pages 21–30, Vancouver, BC, Canada, 2017. ACM Press.

[22] P. R. Pintrich. A conceptual framework for assessing motivation and self-regulated learning in college students. *Educational Psychology Review*, 16(4):385–407, 2004.

[23] E. L. Snow, G. T. Jackson, and D. S. McNamara. Emergent behaviors in computer-based learning environments: Computational signals of catching up. *Computers in Human Behavior*, 41:62–70, Dec 2014.

[24] P. Steel and J. Ferrari. Sex, education and procrastination: an epidemiological study of procrastinators' characteristics from a global sample. *European Journal of Personality*, 27(1):51–58, 2013.

[25] K. M. Styck. Best practices for supporting upward economic and social mobility for first-generation college students. *The School Psychologist*, 72(2):50–57, 2018.

[26] K. Tóth, S. Greiff, C. Kalergi, and S. Wüstenberg. Discovering students' complex problem solving strategies in educational assessment. *Proceedings of the 7th International Conference on Educational Data Mining (EDM)*, pages 225–228, 2014.

[27] J. Whitehill, J. Williams, G. Lopez, C. Coleman, and J. Reich. Beyond prediction: first steps toward automatic intervention in MOOC student stopout. In *Proceedings of the 8nd International Conference on Educational Data Mining*, pages 222–230, Madrid, Spain, 2015.

[28] J. Xie, A. Essa, S. Mojarad, R. S. Baker, K. Shubeck, and X. Hu. Student learning strategies and behaviors to predict success in an online adaptive mathematics tutoring system. In *Proceedings of the 10th International Conference on Educational Data Mining*, pages 460–465, Wuhan, China, 2017.

[29] J. W. You. Examining the effect of academic procrastination on achievement using LMS data in e-learning. *Journal of Educational Technology & Society*, 18(3):64, 2015.

[30] J. W. You. Identifying significant indicators using lms data to predict course achievement in online learning. *The Internet and Higher Education*, 29:23–30, 2016.

[31] B. J. Zimmerman. Self-regulated learning and academic achievement: An overview. *Educational psychologist*, 25(1):3–17, 1990.

Intelligent Instructional Hand Offs

Stephen E. Fancsali
Carnegie Learning, Inc.
501 Grant Street, Suite 1075
Pittsburgh, PA 15219, USA
+1 (412) 992-5099
sfancsali@
carnegielearning.com

Michael V. Yudelson
ACT, Inc./ACT Next
500 ACT Drive
Iowa City, IA 52243, USA
michael.yudelson@act.org

Susan R. Berman
Steven Ritter
Carnegie Learning, Inc.
501 Grant Street, Suite 1075
Pittsburgh, PA, 15219, USA
{sberman, sritter}@
carnegielearning.com

ABSTRACT

Learners in various contemporary settings (e.g., K-12 classrooms, online courses, professional/vocational training) find themselves in situations in which they have access to multiple technology-based learning platforms and often one or more non-technological resources (e.g., human instructors or on-demand human tutors). Instructors, similarly, find themselves in situations in which they can provide learners with a variety of options for instruction, practice, homework, and other activities. We seek data-driven guidance to help facilitate intelligent instructional "hand offs" between learning resources. To begin this work, we focus on an important element of self-regulated learning, namely help seeking. We build classifier models based on proxies for learner prior knowledge and data-driven inferences about learners' disengaged behavior (e.g., gaming the system) and affective states (e.g., confusion) to determine the extent to which (and when) learners tended to seek out help via human tutoring while using an intelligent tutoring system for mathematics. Insights into cognitive, behavioral, and affective factors associated with help seeking outside of a system will drive future work into providing automated, intelligent guidance to both learners and instructors. We close with discussion of the limitations of the present analysis and avenues for future work on intelligently guiding instructional hand offs.

Keywords

intelligent tutoring systems, Cognitive Tutor, mathematics education, developmental mathematics, higher education, online courses, human tutoring, detector models

1. INTRODUCTION

The proliferation of technology-based learning platforms and applications (apps), including intelligent tutoring systems (ITSs), game-based learning environments, massively open online courses (MOOCs), training simulators, language learning apps, and practice apps, among others, creates a complex array of choices for learners and those who would seek to facilitate learning. Far from replacing human instruction, these technologies are often used in learning environments in which learners have access to both technological and human sources of instruction.[1]

Instead of comparing the relative effectiveness of technological and human instruction (c.f. [12, 34]), we are concerned with the extent to which learners' interactions with both technology-based and human resources can be treated as a system that is a target for optimization. One key target for optimizing such a system is the ability to intelligently guide "hand offs" or transitions between different learning apps and to guide learner help seeking as they use technology but also have access to (limited) human resources like an instructor or tutor. Work that considers such hand offs, and intelligent guidance for them (e.g., when a system or app could best provide feedback that directs the learner to an external resource because they need help or could benefit from practice on a pre-requisite skill that is not covered by the system or app), is limited, though one noteworthy exception attempts to provide adaptive assistance as students learn to program by suggesting open, online reading content related to errors made while the student programs [33].

One key element of self-regulated learning [37] is the ability for learners to appropriately and effectively seek out and use help when they need it [3, 27]. ITSs and other technology platforms for learning frequently provide learners with hints and other forms of scaffolding, guidance, and help. Unfortunately, learners often do not make efficient or extensive use of such help within ITSs [1, 25, 36], and when they do, learners sometimes "abuse" such help [2], whether by rapidly seeking progressively more informative hints or attempting to "game the system" [6]. More recent work begins to explore when students *ought* to seek help *within* an ITS. For example, one study found that help avoidance earlier in the problem solving sequence, as students solve genetics problems in an ITS for genetics, is more strongly and negatively associated with robust learning outcomes, suggesting that early help seeking ought to be encouraged [4]. Work like that of [4] is a part of a broader literature focusing on providing meta-cognitive support and developing "meta-cognitive" tutors (e.g., [2]).

Classroom practices in blended, K-12 classrooms also encourage self-regulated learning. Here, students typically have direct access to a teacher while they work within an environment like an ITS. Teachers often adopt strategies like "ask three then me" [17] to

[1] The second author's primary contribution to this work was made while he was employed by Carnegie Learning, Inc., and later Carnegie Mellon University.

encourage productive behavior with respect to help seeking, rather than over-reliance on the teacher. Following this strategy, for example, the student might use the hint feature of an ITS, and should that not provide sufficient clarity or guidance, ask the student on each side of her in the classroom before asking the teacher for help. Given tendencies to over-use and under-use help, better student self-regulation is one important element in optimizing the teacher's scarce time. Ideally, over-users of help will start to rely on help provided by the ITS or their peers, encouraging productive collaboration among learners and enabling teachers to spend more time with students experiencing genuine struggle with content or who rarely seek out help despite needing it.

In the present study, rather than a traditional or blended K-12 classroom, we consider use of the Cognitive Tutor [26] ITS, in one or more of a sequence of two, five-week, fully online developmental mathematics course at a large, mostly-online university. In addition to an instructor, available to students via e-mail and an online message board, students in these courses had optional and unlimited access to human mathematics tutors via a service called Tutor.com (TDC). We were able to obtain access to all chat logs with TDC, as well as detailed data on CT use, providing an ideal dataset to investigate how students navigated between human and automated support in this environment.

In the present study, we focus on cognitive, behavioral, and affective factors that predict whether (and the extent to which) students using CT seek out help from human tutors via an online chat service called Tutor.com. To do so, we adopt a discovery with models approach [10] and build classifier models based on proxies for learner prior knowledge and data-driven inferences about learners' disengaged behavior (e.g., gaming the system, guessing, off-task behavior) and affective states (e.g., confusion, boredom), relying on "detector" models of such factors [5-9]. Insights into cognitive, behavioral, and affective factors associated with help seeking outside of an ITS will drive future work into providing automated, intelligent guidance to both learners and instructors.

2. COGNITIVE TUTOR (CT) & TUTOR.COM (TDC)

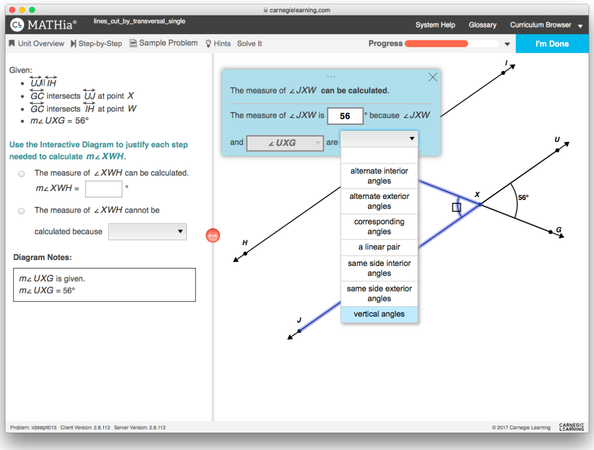
Cognitive Tutor (now called MATHia in K-12 contexts and Mika in higher education contexts) is a mathematics ITS developed and distributed by Carnegie Learning, Inc. [26], used by hundreds of thousands of learners each year in K-12 and higher education learning contexts (see Figure 1).

As illustrated in Figure 1, learners in CT work through complex, multi-step math problems. Within each problem, steps are mapped to fine-grained skills or knowledge components (KCs) [24]. KC mastery is tracked using Bayesian Knowledge Tracing [15].

CT's instructional approach is based on mastery learning [11], and it relies on BKT and these parameters to update estimates of a learner's mastery of the KCs it tracks, as they practice and learn the KCs, within each of its topical sections of content. Within each section, CT presents problems to learners that emphasize the KCs they have yet to master. After mastering all KCs in a section, learners "graduate" to the next section. Having failed to master all of a section's KCs by a certain pre-set limit (e.g., a maximum number of problems), the learner is "promoted" to the following section. MATHia/Mika analytics provide the teacher with information about graduation and promotion status; in promotion cases, teachers will know that the student has failed to master KCs for a particular topic, allowing them to provide some form of remediation, including possibly allowing for a second attempt to work through problems in the ITS later.

As students learn and practice, CT provides context-sensitive, adaptive hints and other feedback. In a typical, blended, K-12 classroom environment in which CT is frequently used, students using CT are in physical proximity to their fellow students and teachers, so they can rely on these resources for help if, for some reason, the CT is not providing sufficient feedback and help. In the present context, CT is used in a fully online context, so for real-time help, the student has to rely on human math tutors, made available to them via an online chat mechanism provided by Tutor.com (TDC). Student could also communicate asynchronously with their course instructors via e-mail and with their fellow students and instructor via an online message board, but data surrounding these means of communication were unavailable to the authors.

TDC is a large provider of online, one-to-one, and on demand tutoring for students in a variety of domains and settings (including learners in K-12 public schools, colleges, universities, libraries, corporations, and the U.S. military). In the context of the present study, TDC tutors were accessible to students, via an online chat mechanism, as a part of their enrollment in the two developmental math courses of which CT was a mandatory instructional component and the primary means by which students were provided with problem-solving practice and exercises. Students were typically assigned several units of content (i.e., sets of sections of content) for each week of the course and allowed, generally, to progress at their own pace through those sections with the expectation that they would complete assigned content within the week in which it was assigned or shortly thereafter.



Given:

- $\overleftrightarrow{UJ} \| \overleftrightarrow{IH}$
- $\overleftrightarrow{GC}$ intersects $\overleftrightarrow{UJ}$ at point X
- $\overleftrightarrow{GC}$ intersects $\overleftrightarrow{IH}$ at point W
- $m\angle UXG = 56°$

Use the Interactive Diagram to justify each step needed to calculate $m\angle XWH$.

○ The measure of $\angle XWH$ can be calculated.

$m\angle XWH =$ □ °

○ The measure of $\angle XWH$ cannot be calculated because □

Diagram Notes:

$m\angle UXG$ is given.
$m\angle UXG = 56°$

The measure of $\angle JXW$ can be calculated.

The measure of $\angle JXW$ is 56 ° because $\angle JXW$ and $\angle UXG$ are

alternate interior angles
alternate exterior angles
corresponding angles
a linear pair
same side interior angles
same side exterior angles
vertical angles
I
U
X
56°
G
H
J
Problem: lcbtstplt015 Client Version: 2.8.113 Server Version: 2.8.113
© 2017 Carnegie Learning
CARNEGIE LEARNING

Figure 1. Cognitive Tutor/MATHia/Mika screenshot

3. DATA

For the present study, the population of concern is comprised of 16,905 adult students in at least one course (and in many cases both courses) in a sequence of two, five-week development mathematics courses for the time period of June 2014 to December 2014, inclusive. Of these 16,905 students over this time period, 80.4% (13,585) made no use of TDC. 3,320 students used TDC at least once during at least one of these courses, with a total number of 19,248 TDC sessions taking place over this six-month period. Tutoring chat sessions lasted from several minutes to over an hour, many occurring while learners simultaneously used CT

Though outside the scope of this study, data available also included transcripts of the TDC tutoring chat sessions (and annotations of dialogue acts [32], instructional modes, and switches between these modes within these chat sessions) that allows for sophisticated analyses of interactions between human and automated tutoring systems like ITSs. These topics, using data from this context, have been explored elsewhere [28-29]. However, data like demographics, student background, and performance in other courses were not available to the authors.

In the analysis that follows, we consider a subset of this population, including 3,119 students who used TDC at least once (i.e., all of the students for whom data could be processed for analysis) as well as a random sample of 1,874 students who did not use TDC over this time period.[2] For these students, we have extensive usage data from CT and rely on the timestamps at which TDC sessions started to identify, for example, the CT login session that occurred before each TDC session. We also know, for each TDC User, the number of times they accessed TDC tutoring sessions as well as the duration of these sessions.

CT data for these 4,993 students were processed into a format amenable to the LearnLab DataShop [20, 23]. These data are comprised of 88,497,091 learner actions (i.e., attempts at steps within problems, or tutor transactions in the DataShop parlance) (an average of 17,724 tutor transactions per student).

The second course in the two-course sequence was more advanced and contains both more challenging content (as measured by CT hints requested and errors made) and fewer sections than the first course in the sequence. Nevertheless, there appear to be few major differences in TDC usage (considering session counts, etc.) between the two courses, so our analyses combine data from the two courses. However, not every student in the sample considered was enrolled in both courses over the time window we consider, so some students only have usage data from the first course and some only from the second course.

4. INITIAL OBSERVATIONS & RESEARCH QUESTIONS

Two related, initial observations inform the analyses of the rest of this work. The first relates to the extent to which a small minority of users accounts for a majority of TDC use. The second observation concerns the imbalance in the data, which informs the overall analytic approach we adopt.

4.1 TDC Super-Users, TDC Users, and TDC Non-Users

Figure 2 provides a histogram counting the number of students with a particular number of TDC sessions. As noted previously, over 13,000 students make no use of TDC and have zero TDC sessions. However, the long right tail of this histogram points to a small minority of students who have tens or even hundreds of TDC sessions. We call students in the top 10% of TDC usage (by session, considering only students with at least one session) "TDC Super-Users." The set of TDC Super-Users is comprised of 350 students (or 2.1% of students in these courses over this period) and account for 55.4% of total TDC session time (4,100 hours of TDC session time). On average, TDC Super-Users spent 7.6 hours in TDC sessions over the period of one of these courses. TDC Users (2,769 students with at least one TDC session but who are not TDC Super-Users) spent a total of 3,367 hours in TDC sessions over this time period, with an average of .8 hours of TDC session time per course.

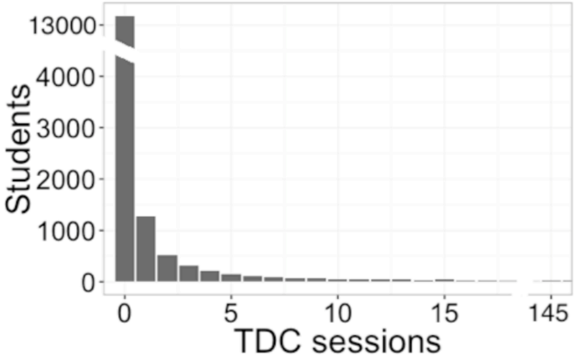


Figure 2. Histogram of TDC sessions and student counts over both courses in the two-course developmental math sequence. Reproduced from [Fancsali, et al unpublished report].

Perhaps unsurprisingly, TDC Super-Users also spent more overall time in CT with an average of 61.7 hours of CT time per course. TDC Users spent an average of 48 hours in CT per course while TDC Non-Users spent only an average of 29.3 hours per course in CT. A more extensive analysis of specific differences and comparisons on various performance metrics for these groups within CT is found in [21].

Such numbers seem likely, though not necessarily[3], to reflect over-use and near-certain under-use (for TDC Non-Users) of the human tutoring provided by TDC. Such over-use and under-use could reflect an underlying problem in terms of self-regulated help seeking. As such, two research questions are directed at the possibility of predicting whether a student is likely to be a TDC Super-User or a TDC User (versus TDC Non-Users). What are possible drivers of such extensive use of TDC? What behaviors and affective states might indicate a need for external help? At a more granular level, the third question seeks to determine whether it is possible to predict from data from a particular CT login session that a student is likely to seek out TDC.

As noted earlier, we center our attention on cognitive features (related to prior preparation for the course), behavioral features like gaming the system, and affective features like boredom,

[2] Seemingly arbitrary counts of 3,119 students who used TDC at least once and the random sample of 1,874 students who never used TDC are largely the result of data collection and data processing limitations in the legacy deployment of Cognitive Tutor used by these students. Some students' data were not reliably collected and/or processed (leading to the difference between 3,320 TDC Users and 3,119 students considered), and time constraints made it impossible to consider a larger sample of TDC Non-Users. Fortunately, present-day implementations of MATHia and Mika no longer suffer from such limitations.

[3] TDC Super-Users, the set of which, for example, could include learners with some form of learning disability like dyscalculia, may derive great learning benefit from interacting with these tutors at the level at which they do (and may need relatively intense remediation to succeed), but this benefit comes at the relatively greater expense of the real-time, chat-based tutor, compared to, for example, regularly setting up time to interact with the instructor or finding other resources for the student to consider when they need such intense help.

among others, that may help to inform future work and provide practical guidance to teachers and facilitators of instruction.

4.2 Research Questions

For each of the following questions, there is a corresponding prediction task for which we consider cognitive, behavioral, and affective factors. Insight provided by predictive models for these tasks (i.e., a better understanding of how prior preparation, disengaged behavior, and affect are associated with seeking human help) is our primary concern in this work. Cognitive factors we consider are related to performance within the first week of a course as a proxy for prior knowledge of the topic and initial effort in the course. Behavioral factors are related to learner disengagement. We detail the features for each prediction task in §5.4.

- What factors predict that a student will be a TDC Super-User? [Prediction Task #1]
- What factors predict that a student will be a TDC Super-User or a TDC User? [Prediction Task #2]
- What factors predict that a particular student login session within CT will be followed by a TDC session? [Prediction Task #3]

For each prediction task, we also consider overall performance metrics for models we describe in §5.3, including accuracy, precision, recall, and AUC to demonstrate the possibility of delivering successful predictive models for these tasks.

We present the tasks in roughly the order of difficulty from easiest to hardest. In the first task, we attempt to distinguish TDC Super-Users from TDC Non-Users, which we *a priori* expect to be an easier task than distinguishing all users of TDC (i.e., the union of the set of TDC Super-Users and TDC Users) from TDC Non-Users. Finally, looking at individual CT login sessions, we seek characteristics of a student's behavior and affect within the CT session itself as well as general characteristics of the student that may predict she is likely to seek out human help.

The predictive models learned for each of these tasks are retrospective (or perhaps descriptive) in the sense that they rely on data aggregated over students' entire usage of Cognitive Tutor in one or both classes for Prediction Tasks #1 and #2 and data from an entire login session for Prediction Task #3. They serve to help direct future studies toward particular factors that might be included in online algorithms or recommendation systems that implement intelligent instructional hand offs (i.e., in real-time, provide a recommendation that it would be conducive to learning for a student to seek out the help of human tutor from TDC, for example, rather than continue to struggle in Cognitive Tutor).

5. METHODS & APPROACH

In this section, we describe our discovery with models approach, using the output of data-driven behavior and affect detectors as input to classifier models to produce predictions for each of our three prediction tasks. We also describe our iterative under-sampling approach to deal with the extent of imbalance present in this dataset.

5.1 Data-Driven Behavior & Affect Detectors

Extensive literature in educational data mining, learning analytics, human computer interaction, and other disciplines focuses on using sensor-free, data-driven approaches for platforms like ITSs to make inferences about student behavior and affect. This literature has produced a wide variety of "detector" models for various behaviors, especially related to disengagement, and affective states for a bevy of learning platforms (e.g., [5-9, 22, 31, 35]).

In this work, we rely on detectors of disengaged behavior and affect while students use CT. Detectors were implemented for gaming the system [7], off-task behavior [9], and affective states including: boredom, confusion, frustration and engaged concentration [8]. In addition, we implemented contextual models of guessing and slipping to estimate the extent to which each may have been responsible for correct and incorrect answers (i.e., estimating when it may be likely that students are guessing correctly without KC mastery and slipping to produce an incorrect answer despite mastery of a KC) [5]. Contextual slip models have been used as detectors of carelessness in previous work [19, 31]. Gaming the system [6] refers to behavior directed at making progress through content without genuine learning. Learners may try to make progress by adopting strategies like relying on "bottom out" hints that provide the answer or by providing numbers that appear within problem statements as answers to questions, among other shallow (at best) learning strategies.

Detectors we deploy in this study have been successfully used with a similar population of learners in previous work [18-19]. Detectors of gaming the system, off-task behavior, and models of contextual guessing and slipping produce predictions at the level of individual learner actions (i.e., attempts at problem-solving-steps) while detectors of affective states produce predictions about "clips" or time intervals of approximately 20 seconds. For a more extensive summary of the features that are "distilled" from CT log data to serve as input to the underlying machine learning models that constitute these detectors, please see papers cited for each detector [7-9] as well as the papers describing their use with a similar population of higher education CT learners [18-19].

5.2 Imbalanced Data & General Approach

For each prediction task, we adopt an iterative scheme to deal with the fact that each task involves imbalanced data in terms of the target of predictive interest. While a variety of approaches are amenable to the task of dealing with imbalanced data, in the present study, we are primarily interested in establishing the characteristics of disengaged behaviors, affective states, and prior knowledge that predict that students seek out human help, so we adopt a strategy of iteratively considering balanced samples of data, building classifier models on these balanced samples, and considering the factors that contribute to the success of these classifiers. For Prediction Task #1, there are 350 TDC Super-Users and 1,874 TDC Non-Users. For Prediction Task #2, there are 3,119 TDC Super-Users and TDC Users and 1,874 TDC Non-Users. For Prediction Task #3, 3,058 of the 3,119 TDC Super-Users and TDC Users have at least one CT login session that is followed by a session with a TDC tutor while there are 580,528 CT login sessions overall.[4]

For each prediction task, we create a balanced sample by under-sampling the appropriate majority class in each of 500 iterations, building classifier models in each. For Prediction Task #1, this means creating (500x) a sample (with student-level features we describe in §5.4) of the same 350 TDC Super-Users and a random sample of 350 TDC Non-Users. For Prediction Task #2, we create a sample (again, 500x, with student-level features) containing the

[4] 61 students use TDC one or more times before using CT in the courses, so there are no CT sessions from which data can used to better understand what predicts that student's decision to use a TDC tutor.

same 1,874 TDC Non-Users and a random sample of 1,874 students drawn from TDC Users and TDC Super-Users. For Prediction Task #3, we randomly sample one CT session per student that is followed by a TDC session and randomly sample one CT session per student (also chosen at random) that is not followed by a TDC session, resulting in a sample of 6,116 CT sessions for which we have CT session-level features we describe in §5.4. This approach for Prediction Task #3 avoids violations of independence that would be introduced by students with multiple TDC sessions were we to consider more than one session per student.

In each iteration, we have a balanced dataset of student-level or CT-student-session-level features that can be used as predictors in classifier models. We take a 60%-40% split of this dataset into training and test sets, and build classifier models using 5-fold cross validation on the training set, which, given the way we have constructed the training and test set, is student-stratified cross validation. We apply the best performing model in terms of accuracy over this 5-fold cross validation to the held-out test set. Having done this process 500x for each prediction task, we consider the mean (and standard deviation of) performance over these iterations using metrics of accuracy, precision, recall, and area under the ROC curve (AUC). We also consider the specifics of a representative model for each prediction task to provide insights into which features are predictive of seeking out human help.

To test the robustness of this approach, for the case of Prediction Task #2, which is not drastically imbalanced (i.e., 37.5% of students in the sample are non-TDC Users), we consider models learned without using this iterative under-sampling scheme. We show that results are comparable in terms of AUC and compare other performance metrics between the approach, helping to establish possible bounds on expected predictive accuracy and other metrics. Classification accuracy, for example, in this under-sampling scheme is perhaps an especially optimistic estimate of what can be achieved.

5.3 Classifier Models

We consider four types of models to drive classification and prediction: logistic regression (LR), random forest (RF) [13], and support vector machines [16] with both linear (SVML) and radial kernels (SVMR). For each model, we consider the case in which the models output binary classifications as well as probabilities for each of the binary classes of the target variable. In this way, we are able to consider classification accuracy, precision, and recall, as well as AUC as a further comparison of performance compared to chance. Estimated LR models provide a convenient way to consider the significance of features included in these models, so we illustrate the importance of variables in these models in this way.

5.4 Feature Construction

For Prediction Tasks #1 and #2, student-level features are constructed over usage for the entire period of time over the courses in which each student had usage (either the first course, second course, or both). Such features provide for a general profile of how students worked through content in these two courses. Features represent predictions made by detector models as previously described as well as variables related to student performance and usage in their first week of CT usage in the first course they encountered (if they used CT in both courses). Features constructed from "Week 1" data are proxies, however noisy, for student prior preparation and initial knowledge, as other measures, as previously noted, were unavailable. Each variable is constructed as a normalized z-score over all students in the dataset (i.e., the unit for each variable is the number of standard deviations above or below the mean value for each feature):

- *Assistance Per Step*: Mean number of hints requested + errors per problem-solving step
- *Gaming the System*: Proportion of student actions inferred to be instances of gaming the system behavior.
- *Off-Task*: Proportion of student actions inferred to be instances of off-task behavior.
- *Guessing*: Proportion of correct student actions inferred to be possible instances of having correctly guessed.
- *Slipping*: Proportion of incorrect student actions inferred to be possible instances of having slipped despite KC mastery.
- *Boredom*: Proportion of problem solving clips in which students were judged by detector models to have been bored.
- *Frustration*: Proportion of problem solving clips in which students were judged by detector models to have been frustrated.
- *Confusion*: Proportion of problem solving clips in which students were judged by detector models to have been confused.
- *Engaged Concentration*: Proportion of problem solving clips in which students were judged by detector models to be in a state of engaged concentration.
- *Week 1 Sections*: Number of sections of content encountered in the first week of either course (or across both).
- *Week 1 Assistance*: Hints requested and errors made in the first week of either course (or across both).
- *Week 1 Time*: Amount of time spent using Cognitive Tutor in the first week of either course (or across both).
- *Week 1 Sections/Hour*: Number of sections of content encountered per hour in the first week of either course (or across both).
- *Week 1 Assistance/Hour*: Number of hints requested and errors made per hour in the first week of either course (or across both).
- *Week 1 Completer*: binary indicator that a student encountered 90% of the sections in the first week's assignment in either course (or across both).

For Prediction Task #3, CT login-session level features are considered. These features are not normalized, but rather the same proportions as for Prediction Tasks #1 and #2 but with respect to a particular CT login session. For example, Assistance Per Step is calculated over only problem-solving steps within a CT session. *Gaming the System* is calculated as the proportion of student actions within a CT login session that are predicted to be instances of gaming the system, and *Boredom* is calculated as the proportion of problem-solving clips within a CT login session for which detector models infer that a student is bored. Week 1, student-level variables are also included in these models.

6. RESULTS

For each prediction task, we first describe the predictive performance for each of the models we deploy, and then we consider a "representative" logistic regression model that provides insight into the factors that help us to achieve success on these tasks. We describe the sense in which we consider these logistic regression models to be "representative" in the following sub-section.

6.1 Prediction Task 1: TDC Super-Users

We expect the task of distinguishing TDC Super-Users from TDC Non-Users to be the "easiest" *a priori*, in the sense that we expect that we will be able to achieve better performance on the task, an expectation which is borne out by our results. Table 1 shows that logistic regression (LR) performs comparably to a support vector machine with a linear kernel (SVML) with mean accuracy over 500 iterations of .712 and nearly identical values for precision, recall, and AUC. Recall that .5 accuracy represents chance accuracy (and .5 AUC represents chance performance, as ever) because we under-sample to produce a balanced dataset in each iteration.

Table 1. Mean and standard deviation (in parentheses) for accuracy, precision, recall, and area under the ROC curve (AUC) over 500 iterations for the task of predicting whether a student is a TDC Super-User (versus a non-TDC User) [LR = Logistic Regression; RF = Random Forest; SVML = Support Vector Machine with Linear Kernel; SVMR = Support Vector Machine with Radial Kernel]

Model	Accuracy	Precision	Recall	AUC
LR	.712 (.025)	.701 (.029)	.744 (.042)	.786 (.024)
RF	.705 (.025)	.698 (.028)	.727 (.042)	.771 (.024)
SVML	.712 (.024)	.702 (.03)	.743 (.048)	.788 (.023)
SVMR	.665 (.025)	.65 (.03)	.7245 (.056)	.723 (.027)

Table 2 provides a representative, estimated logistic regression model that provides insight into student-level factors that are associated with a student being a TDC Super-User. The model is representative in the sense that, upon inspection of multiple models built on training sets sampled in the way we described above, the significant variables in the model of Table 2 were generally those that were significant. We then specified logistic regression models including only the variables that are reported significant in Table 6 and found that these models, over hundreds of iterations, achieved results nearly identical to those reported for logistic regression in Table 1. Spot inspections of model parameters in numerous models produced by the iterative process also aligned with those reported in Table 2 in terms of both sign and magnitude. This same notion of representative logistic regression models is used for each of the three predictive tasks we consider to provide insight into the variables that contribute to such models.

The model of Table 2 suggests that the four significant factors for predicting that a student will be a TDC Super-User are *Off-Task* disengagement, *Boredom*, *Guessing,* and *Week 1 Sections/Hour*. Pairwise Pearson correlations among these significant predictors are small, with no statistically significant correlation between *Guessing* and *Off-Task* disengagement, and the largest significant correlation is that between *Week 1 Sections/Hour* and *Boredom* (r = .36; p < .001). These observations, combined with the consistency of models learned over only these significant predictors, instill confidence in our interpretation of the logistic regression coefficients. However, multi-collinearity among some of the other predictors (especially, for example, *Gaming the System* and *Confusion*: r = .76; p < .001) requires us to exercise caution in interpreting other coefficients in this representative logistic regression model. Roughly these same observations about the significant predictors as well as caveats concerning the interpretation of the non-significant estimated regression coefficients are operative for Predictive Tasks #2 and #3.

While disengagement is positively associated with TDC Super-User status, the *Boredom*, *Guessing*, and *Week 1 Sections/Hour* are negatively associated with TDC Super-User status, indicating that students who are inferred to be less bored, less likely to be haphazardly guessing, and better prepared for the coursework (as indicated by efficient progress through content in the first week of the course) are less likely to seek out human help extensively.

Table 2. Representative estimated logistic regression model for the task of predicting whether a student is a TDC Super-User (versus a non-TDC User). ***Rows for significant variables at α = 0.05 are bold and italicized.***

Variable	Coefficient	Std. Error	p-value
(Intercept)	-.756	.604	.21
Assistance Per Step	.664	.526	.207
Gaming the System	.103	.271	.704
Off-Task	***.35***	***.161***	***.03***
Guessing	***-.611***	***.306***	***.046***
Slipping	.135	.17	.429
Boredom	***-.774***	***.292***	***.009***
Frustration	.249	.182	.172
Confusion	-.084	.233	.72
Engaged Concentration	.376	.306	.218
Week 1 Sections	.361	.193	.061
Week 1 Assistance	-.333	.322	.302
Week 1 Time	.058	.277	.833
Week 1 Sections/Hour	***-1.365***	***.425***	***.001***
Week 1 Assistance/Hour	.052	.285	.856
Week 1 Completer	.478	.636	.452

6.2 Prediction Task 2: TDC Users + TDC Super Users

As expected, we find that distinguishing those students who used TDC at least once (the set of TDC Users + TDC Super-Users) from TDC Non-Users is more "difficult" in the sense that models achieve a lower degree of classification accuracy, precision, and recall, as well as a lower AUC (Table 3).

Inspection of the representative, estimated LR model in Table 4 indicates that in addition to the three same features that are significant in predicting TDC Super-User status (i.e., *Off-Task* disengagement, *Boredom*, and *Guessing*), *Week 1 Time* is a significant predictors that students will have used TDC at least once, suggesting that this measure of time provides different information to help distinguish between these categories of students.

Table 3. Mean and standard deviation (in parentheses) for accuracy, precision, recall, and area under the ROC curve (AUC) over 500 iterations for the task of predicting whether a student used TDC at least once (i.e., TDC Super-User or TDC User versus a non-TDC User) [see model acronyms in caption for Table 1]

Model	Accuracy	Precision	Recall	AUC
LR	.614 (.0111)	.62 (.012)	.592 (.023)	.666 (.012)
RF	.615 (.0109)	.614 (.0115)	.618 (.0209)	.66 (.0113)
SVML	.612 (.0105)	.624 (.0133)	.5651 (.0377)	.6656 (.0113)
SVMR	.598 (.0115)	.6 (.013)	.591 (.0332)	.629 (.0121)

Table 4. Representative estimated logistic regression model for the task of predicting whether a student is a TDC User (versus a non-TDC User). ***Rows for significant variables at α = 0.05 are bold and italicized.***

Variable	Coefficient	Std. Error	p-value
(Intercept)	-.1	.2	.617
Assistance Per Step	-.068	.118	.564
Gaming the System	-.05	.092	.586
Off-Task	***.133***	***.063***	***.035***
Guessing	***-.233***	***.071***	***< .001***
Slipping	-.021	.056	.706
Boredom	***-.527***	***.088***	***< .001***
Frustration	.067	.045	.142
Confusion	.132	.092	.149
Engaged Concentration	.042	.096	.661
Week 1 Sections	-.117	.064	.067
Week 1 Assistance	-.226	.115	.05
Week 1 Time	***.378***	***.128***	***.003***
Week 1 Sections/Hour	-.135	.077	.08
Week 1 Assistance/Hour	-.085	.081	.29
Week 1 Completer	.205	.213	.335

Since this prediction task is the least imbalanced of the three we consider, we also consider learning models without our adopted under-sampling scheme. Though we omit extensive analysis of these models for brevity, Table 5 provides performance metrics for LR and RF models learned by taking a 60-40% training-test split of all students, learning models using 10-fold cross validation on the training set and applying the model with greatest accuracy to the test set. We find that this model modestly out-performs the trivial, majority class classifier in terms of classification accuracy with comparable precision, but recall of this model is substantially greater than that achieved by typical models in our under-sampling scheme.

Building on our observations from the previous model, since *Week 1 Time* has a positive parameter estimate, students who take more time to work through content in the first week, and perhaps work more diligently by guessing less as they make problem-solving attempts, may be more likely to seek out help via TDC. It is possible that otherwise relatively diligent students (by some measures) who seek out TDC begin to adopt a sub-optimal learning strategy of some sort that is indicated by the *Off-Task* detector more frequently than those students who do not seek out TDC.

Consequently, the F measure (one commonly used evaluation metric that balances precision and recall) would be greater for these models than for those of the typical models of our under-sampling scheme. Nevertheless, AUC of these models are nearly identical to mean values of models learned according to our under-sampling scheme. Perhaps more importantly, the estimated logistic regression model points to exactly the same set of significant behavioral and affective features, *Off-Task* disengagement, *Boredom*, and *Guessing*, as the model reported in Table 4. *Week 1 Time* is also significant in models using both approaches, though *Week 1 Sections*, *Week 1 Assistance/Hour*, and *Week 1 Completer* are significant in the model that does not rely on under-sampling.

Table 5. Accuracy, precision, recall, and area under the ROC curve (AUC) for the task of predicting whether a student used TDC at least once (versus a TDC Non-User) for models estimated without relying on under-sampling scheme (trivial majority classifier accuracy = .625)

Model	Accuracy	Precision	Recall	AUC
LR	.666	.684	.865	.669
RF	.651	.68	.832	.659

6.3 Prediction Task 3: TDC Sessions Follows a CT Login Session

As expected, the most difficult task was to predict whether a particular CT session was going to be followed by a TDC session, as illustrated by the performance metrics for the various models we consider in Table 6.

Table 6. Mean and standard deviation (in parentheses) for accuracy, precision, recall, and area under the ROC curve (AUC) over 500 iterations for the task of predicting whether a particular student CT session is followed by a session with a TDC tutor [see model acronyms in caption for Table 1]

Model	Accuracy	Precision	Recall	AUC
LR	.599 (.009)	.604 (.015)	.579 (.021)	.633 (.01)
RF	.587 (.01)	.587 (.014)	.592 (.023)	.621 (.011)
SVML	0.6 (.009)	.61 (.015)	.559 (.023)	.633 (.01)
SVMR	.598 (.01)	.606 (.017)	.567 (.031)	.632 (.01)

Table 7 provides a representative, estimated LR model that provides insight into the factors that are predictive of a student's tendency to seek out human tutoring via TDC from within a particular CT session. Here, *Boredom* appears again, along with *Engaged Concentration* (which was significant in neither Prediction Task #1 nor Prediction Task #2), as a significant, negatively associated predictor. We also find that *Gaming the System*, another form of disengagement, is positively associated with a tendency to seek out immediate help via TDC, along with *Week 1 Sections*.

At the level of student-login sessions in Prediction Task #3, *Gaming the System* and the other detected factors are no longer highly correlated (as they were when we considered student-level aggregated features in Prediction Tasks #1 and #2). Rather *# Hints* and *# Errors* and *Week 1 Time* and *Week 1 Assistance* are relatively highly correlated, leading us to exercise caution in the interpretation of estimated coefficients associated with these (insignificant) predictors.

Table 7. Representative estimated logistic regression model for the task of predicting whether a particular student CT login session is followed by a session with a TDC tutor. Coefficients are un-standardized. *Rows for significant variables at α = 0.05 are bold and italicized.*

Variable	Coefficient	Std. Error	p-value
(Intercept)	***1.321***	***.364***	***< .001***
# Errors	-.002	.001	.177
# Hints	.002	.001	.262
Gaming the System	***1.367***	***.23***	***< .001***
Off-Task	.612	.652	.348
Guessing	-.718	1.207	.552
Slipping	-.197	.491	.689
Boredom	***-.783***	***.149***	***< .001***
Frustration	.12	.316	.705
Confusion	.213	1.32	.872
Engaged Concentration	***-1.575***	***.229***	***< .001***
Week 1 Sections	***.021***	***.006***	***< .001***
Week 1 Assistance	-.0001	.0001	.179
Week 1 Time	.003	.01	.733
Week 1 Sections/Hour	-.021	.019	.263
Week 1 Assistance/Hour	-.0004	.001	.68

7. DISCUSSION

7.1 Highlights & Summary

At least two qualitative findings are robust in the modeling presented. First, as inferred by detector models in CT, *Boredom* is negatively associated with a tendency to seek out human help outside of the CT ITS via the TDC service in both the aggregate (Prediction Tasks #1 and #2) as well as the more immediate term (Prediction Task #3). Especially when combined with the negative association of *Guessing* with seeking out TDC's services in Prediction Tasks #1 and #2, this suggests at least a modicum of baseline diligence in working within CT for those who sought out TDC. However, the second robust finding may point to the adoption of counter-productive strategies that may also lead students to require assistance outside of the ITS. This second robust finding is that two facets of learner disengagement inferred by such detector models, *Off-Task* behavior and *Gaming the System*, are positively associated, in the aggregate and more immediately, respectively, with seeking human assistance outside of the CT ITS. These insights contribute to a bevy of literature concerning various aspects of the technology-enhanced learning experience, generally centered on learning outcomes and learners using ITSs, which are associated with these phenomena (e.g., [14, 18, 30]).

7.2 Limitations

While we consider a rich, substantial data set with thousands of learners, the present analysis is not without its limitations. First, we merely consider learning models to predict that a student is likely to be particular "type" of TDC user or that a particular CT login session is likely to be followed by a session with a TDC tutor. We do not consider the effectiveness of TDC sessions, though some work has begun to consider that question [28-29], or attempt to deeply link the specific KCs within CT on which students may have been working when they sought out TDC. This dataset also offers the opportunity to consider CT usage and performance (possibly at the level of fine-grained KCs) before and after a TDC session as a type of pre- and post-test for these sessions.

Further, this is a purely retrospective, observational study, and the empirical frequencies with which students sought out (and did not seek out) help via the TDC service reflects likely over-use and near certain under-use. While the models we have learned have provided insights into the context in which these data were collected, data from scenarios and contexts in which we suspect that such use of human tutors is more attuned to need would provide interesting contrast cases to the present study. In addition, while associations uncovered by predictive models like those presented could arise due to causal relationships between factors captured by these predictors, the present analysis does not provide us evidence for any such claims. While adopting counter-productive strategies like gaming the system in CT may precede seeking out human help, is such a counter-productive strategy really the cause of seeking such help? If we were to conceive of a clever intervention to reduce gaming the system behavior, would that reduce the incidence of learners seeking out human help? Future work might more carefully observe students in environments in which they can seek out human help while using an ITS (or other systems) to elicit their explanations for help seeking, or experimental studies might consider interventions that tend to increase or decrease the extent to which students rely on external help.

8. FUTURE WORK

In addition to several opportunities noted in the previous section, we consider two "big" ideas with respect to future work.

8.1 Information vs. Affirmation

One concern with this analysis is that we are building models that combine different motives for students to seek out human assistance. Consider the following dialog (a slightly edited TDC interaction):

> **Tutor**: hi! what can I help you with today?
>
> **Student**: Do you know how to do a factor table?
>
> **Tutor**: Hmm I am familiar with it. Is there a problem that you wanted to go over?
>
> **Student** This looks like an easy one, but I am not sure so I just want to make sure I understand this correctly
>
> **Student:** To check this table is all you do multiply the top row by the 7x and see if it matches the bottom row? Is this right?
>
> **Tutor**: Yeah everything looks good to me**.** Great job!
>
> **Student:** I was hoping that I did this right.

We call this kind of interaction a request for "affirmation," rather than information. The tutor is not teaching the student anything, just verifying that the student's approach is correct. The conditions leading to this type of interaction are likely to be very different from information requests. They may occur when students have high knowledge but low confidence, for example. Future work will explore models that separate information from affirmation sessions.

8.2 Instructional Hand Offs

In contemporary K-12 classrooms, online courses, and other settings for learning, students may seek instruction, assistance, remediation, opportunities for enrichment, and even affirmation from multiple resources, including technology resources like ITSs and non-technological resources like human beings. Especially when at least one of these resources is technological, providing adaptive, intelligent guidance to learners as to when they should use particular resources and applications (or persist and try to "stick with it" and learn within a particular application) will be crucial. In the present study, we have sought to better understand cognitive, behavioral, and affective factors that predict that a student may seek help from a non-ITS resource like a human tutor while using the CT ITS for math, but other types of instructional hand offs should also be considered.

Hand offs between instructional applications might happen, for example, between an ITS and a simulation-based training environment. When a student has completed all of the skills for which the ITS provides instruction, the simulation-based training environment that includes some overlapping content with the ITS could tailor its simulated scenarios around emphasizing elements of those skills in the ITS on which the student struggled and de-emphasize skills that the student easily mastered within the ITS. This is likely to require a *lingua franca* shared by the ITS and the simulator about the competencies or skills that are tracked by each, or perhaps both may rely on a set of external standards or some other way of indicating how this type of hand off based on such cognitive factors may work. Efforts including the development of the Experience API[5] (xAPI), the Total Learning Architecture[6] (TLA), and the Generalized Intelligent Framework for Tutoring[7] (GIFT) exemplify moves in directions that would enable progress toward these and similar goals.

Of course, even in the case of guiding an instructional hand off between an ITS and a human tutor (or K-12 classroom teacher) for a student who needs help with content covered by the ITS, the ITS ideally should be able to communicate to the human tutor or classroom teacher that the learner in question requires assistance on a particular skill, just needs a confidence boost, or has been adopting counter-productive and/or disengaged learning strategies like gaming the system that should probably be discouraged. Insights into predictors of help seeking may help to drive development of recommendations delivered by the learning application to the learner or could also drive recommendations to a teacher via an application that surfaces insights from the ITS.

We hope this work provides a step toward more work on these, and related, problems.

9. ACKNOWLEDGMENTS

This work was funded by a contract from the U.S. Department of Defense Advanced Distributed Learning Initiative (Contract PAAIDT W911QY-14-C-0019). Anonymous reviewers provided helpful comments that have improved the presentation of this work.

10. REFERENCES

[1] Aleven, V., and Koedinger, K. R. 2000. Limitations of student control: Do students know when they need help? In *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, (Montreal, Canada, June 19-23, 2000). ITS 2000. Springer-Verlag, Berlin, 292-303.

[2] Aleven, V., McLaren, B., Roll, I., and Koedinger, K. 2006. Toward meta-cognitive tutoring: A model of help seeking with a cognitive tutor. *International Journal of Artificial Intelligence and Education* 16, 2, 101-128.

[3] Aleven, V. Stahl, E., Schworm, S. Fischer, F. and Wallace, R. 2003. Help seeking and help design in interactive learning environments. *Rev. Educ. Res.* 73, 3, 277-320.

[4] Almeda, V., Baker, R., and Corbett, A. 2017. Help avoidance: When students should seek help, and the consequences of failing to do so. *Teach. Coll. Rec.* 117, 3, 1-24.

[5] Baker, R.S., Corbett, A.T., and Aleven, V. 2008. More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian Knowledge Tracing. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems,* (Montreal, Canada, 2008). ITS 2008. 406-415

[6] Baker, R.S., Corbett, A.T., Koedinger, K.R., and Wagner, A.Z. 2004. Off-task behavior in the Cognitive Tutor classroom: when students "game the system." In *Proceedings of ACM CHI 2004: Computer-Human Interaction* (Vienna, Austria, 2004). 383-390.

[7] Baker, R.S., and de Carvalho, A. M. J. A. 2008. Labeling student behavior faster and more precisely with text replays. In *Proceedings of the 1st International Conference on Educational Data Mining* (Montreal, 2008). 38-47.

[8] Baker, R.S., 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. In *Proceedings of the 5th International Conference on Educational Data Mining*, (Chania, Greece, 2012). 126-133.

[5] https://github.com/adlnet/xAPI-Spec

[6] https://www.adlnet.gov/tla/

[7] https://www.gifttutoring.org/

[9] Baker, R.S. 2007. Modeling and understanding students' off-task behavior in intelligent tutoring systems. In *Proceedings of ACM CHI 2007: Computer-Human Interaction* (San Jose, CA, April 28 – May 3, 2007). ACM, New York, 1059-1068.

[10] Baker, R.S., and Yacef, K. 2009. The state of educational data mining in 2009: a review and future visions. *Journal of Educational Data Mining* 1, 3-17.

[11] Bloom, B. S. 1968. Learning for mastery. *Evaluation Comment* 1, 2, 1–12.

[12] Bloom, B. 1984. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educ. Researcher* 13, 6, 4-16.

[13] Breiman, L. 2001. Random forests. *Mach. Learn.* 45, 1, 5-32.

[14] Cocea, M., Hershkovitz, A., Baker, R.S.J.d. 2009. The impact of off-task and gaming behavior on learning: immediate or aggregate? In *Proceedings of the 14th International Conference on Artificial Intelligence in Education* (Brighton, UK, 2009). 507-514

[15] Corbett, A.T., and Anderson, J.R. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model. User-Adap.* 4, 253-278.

[16] Cortes, C., and Vapnik, V.N. 1995. Support-vector networks. *Mach. Learn.* 20, 3, 273–297.

[17] Daniels, C., Edwards, R., Miller, P., Hale, A., Powell, L., Wisner, J., Mallonee, K., Perkins, S., Bravo, J., Hummel, M., Wagner, B., and Fay, M. 2010. *PowerTeaching: Cooperative Learning Handbook.* Success For All Foundation, Baltimore, MD.

[18] Fancsali, S.E. 2014. Causal discovery with models: behavior, affect, and learning in Cognitive Tutor Algebra. In *Proceedings of the 7th International Conference on Educational Data Mining,* (London, UK, 2014). EDM 2014. 28-35.

[19] Fancsali, S.E. 2015. Confounding carelessness? Exploring causal relationships between carelessness, affect, behavior, and learning in Cognitive Tutor Algebra. In Proceedings of the 8th International Conference on Educational Data Mining, (Madrid, Spain, 2015). EDM 2015. 508-511.

[20] Fancsali, S.E., Ritter, S., Berman, S.R., Yudelson, M., Rus, V., and Morrison, D.M. 2016. Toward integrating Cognitive Tutor interaction data with human tutoring text dialogue data in LearnSphere. In *Proceedings of the EDM 2016 Workshops and Tutorials*, (Raleigh, NC, 2016). CEUR Workshop Proceedings.

[21] Fancsali, S.E., Rus, V., Ritter, S., and Berman, S.R. 2017. *Final Technical Report: Integrating Human and Automated Tutoring Systems*. Technical Report. Carnegie Learning, Inc., Pittsburgh, PA.

[22] Johns, J. and Woolf, B. 2006. A dynamic mixture model to detect student motivation and proficiency. In *Proceedings of the 21st National Conference on Artificial Intelligence* (Boston, MA, 2006). AAAI Press, Menlo Park, CA, 2-8.

[23] Koedinger, K.R., Baker, R.S.J.d., Cunningham, K., Skogsholm, A., Leber, B., and 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, and R.S.J.d. Baker, Eds. CRC, Boca Raton, FL, 43-55.

[24] Koedinger, K.R., Corbett, A.T., and Perfetti, C. 2012. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Sci.* 36, 757-798.

[25] Renkl, A. 2002. Learning from worked-out examples: Instructional explanations supplement self-explanations. *Learn. Instr.* 12, 529-556.

[26] Ritter, S., Anderson, J.R., Koedinger, K.R., and Corbett, A.T. 2007. Cognitive Tutor: applied research in mathematics education. *Psychon. B. Rev.* 14, 249-255.

[27] Roll, I., Aleven, V., McLaren, B.M., Koedinger, K.R. 2011. Improving students' help seeking skills using metacognitive feedback in an intelligent tutoring system. *Learn. Instr.* 21, 267-280.

[28] Rus, V., Banjade, R., Maharjan, N., Morrison, D., Ritter, S., and Yudelson, M. 2016. Preliminary results on dialogue act classification in chatbased online tutorial dialogues. In *Proceedings of the 9th International Conference on Educational Data Mining*, (Raleigh, NC, 2016). EDM 2016. 630-631.

[29] Rus, V., Maharjan, N., Tamang, L.J., Yudelson, M., Berman, S., Fancsali, S.E., and Ritter, S. 2017. An analysis of human tutors' actions in tutorial dialogues. In *Proceedings of the 30th International Florida Artificial Intelligence Research Society Conference*, (Marco Island, FL, 2017). FLAIRS-30. 122-127.

[30] San Pedro, M.O.C.Z., Baker, R.S., Bowers, A.J., and 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* (Memphis, TN). EDM 2013. 177-184

[31] San Pedro, M.O.C.Z., Baker, R. S., Rodrigo, M. 2011. Detecting carelessness through contextual estimation of slip probabilities among students using an intelligent tutor for mathematics. In *Proceedings of 15th International Conference on Artificial Intelligence in Education* (Auckland, New Zealand). 304-311.

[32] Searle, J.R. (1969). *Speech Acts*. Cambridge UP.

[33] Sosnovsky, S. 2011. Ontology-Based Open-Corpus Personalization for E-Learning. Doctoral Thesis. University of Pittsburgh.

[34] vanLehn, K. 2011. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educ. Psychol.* 6, 4, 197-221.

[35] Walonoski, J., and Heffernan, N.T. 2006. Prevention of off-task gaming behavior in intelligent tutoring systems. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, (Jhongli, Taiwan, 2006). ITS 2006. 722-724.

[36] Wood, H., and Wood, D. 1999. Help seeking, learning and contingent tutoring. *Comput. Educ.* 33, 153-169.

[37] Zimmerman, B. J. 1990. Self-regulated learning and academic achievement: An overview. *Educ. Psychol.* 25, 1, 3-17.

Improving Stealth Assessment in Game-based Learning with LSTM-based Analytics

Bita Akram
North Carolina State University
Raleigh, NC 27695
bakram@ncsu.edu

Wookhee Min
North Carolina State University
Raleigh, NC 27695
wmin@ncsu.edu

Eric Wiebe
North Carolina State University
Raleigh, NC 27695
wiebe@ncsu.edu

Bradford Mott
North Carolina State University
Raleigh, NC 27695
bwmott@ncsu.edu

Kristy Elizabeth Boyer
University of Florida
Gainesville, FL 32611
keboyer@ufl.edu

James Lester
North Carolina State University
Raleigh, NC 27695
lester@ncsu.edu

ABSTRACT

A key affordance of game-based learning environments is their potential to unobtrusively assess student learning without interfering with gameplay. In this paper, we introduce a temporal analytics framework for stealth assessment that analyzes students' problem-solving strategies. The strategy-based temporal analytic framework uses long short-term memory network-based evidence models and clusters sequences of students' problem-solving behaviors across consecutive tasks. We investigate this strategy-based temporal analytics framework on a dataset of problem-solving behaviors collected from student interactions with a game-based learning environment for middle school computational thinking. The results of an evaluation indicate that the strategy-based temporal analytics framework significantly outperforms competitive baseline models with respect to stealth assessment predictive accuracy.

Keywords

Game-based Learning, Stealth Assessment, Temporal Analytics, LSTM, Strategy Use

1. INTRODUCTION

Recent years have seen significant growth in investigations of game-based learning. Game-based learning environments utilize the motivational elements of games to foster students' learning and engagement [7, 34, 36]. Studies have shown that learners who engage in game-based learning experience higher motivation compared to those who learn with conventional methods [8, 39]. Intelligent game-based learning environments integrate the adaptive learning support of intelligent tutoring systems and the motivational elements of games [15]. Like intelligent tutoring systems, they utilize students' interactions with the learning environment to infer student models of cognitive, affective, and metacognitive states [20, 26, 40]. The resulting student models can then guide tailored problem-solving scenarios, cognitive feedback, affective support [1,25].

In contrast to traditional assessment, stealth assessment of student learning can rely solely on student interaction trace data from the game-based learning environment without disrupting the natural flow of learning [38]. Stealth assessment infers students' competency with respect to knowledge, skills, and performance using evidence derived from students' game-based learning activities often based on evidence-centered design (ECD) [27]. ECD utilizes task, evidence, and competency models to assess students' relevant competency and proficiency [35]. In game-based learning environments, stealth assessment can monitor granular game-based behaviors across multiple tasks in the game to generate evidence, which can then be used to dynamically infer a competency model of the student. Operating in this fashion, stealth assessment has been examined to unobtrusively perform assessments of a wide range of constructs [40], and provide formative feedback to students and teachers to inform instruction and enhance learning [39, 5, 18].

Although an abundance of data can be readily captured from student interactions within game-based learning environments, a key challenge posed by stealth assessment is translating the raw data into meaningful representations to model students' competencies and performance [39]. This problem is exacerbated by the fact that student behavior unfolds over time in a manner dependent on prior actions. In this work, we present an approach to stealth assessment that leverages temporal analytics based on students' problem-solving strategies. Building on findings that problem-solving strategies significantly influence learning outcomes [11, 33], we introduce a strategy-based temporal analytics method using *n*-gram features and investigate whether problem-solving strategies identified from clustering students' interaction patterns can improve the predictive accuracy of evidence models for stealth assessment.

After clustering students based on their problem-solving behaviors, we predict their post-test performance using their cluster assignments as predictive features for a suite of classifiers. This approach is based on the intuition that as students' progress through a series of learning tasks, their choice of strategy affects their learning outcomes. For example, if a student first pursues a trial-and-error strategy for initial tasks and later in the learning session begins to adopt a more effective strategy, her strategy shift may lead to higher post-test scores. We hypothesize that drawing inferences about strategy shifts may serve as the basis for accurate predictions of learning performance.

Because strategies and strategy shifts are inherently time-based phenomena, we propose a strategy-based temporal analytics approach to stealth assessment based on long short-term memory networks (LSTMs). In this approach, we develop predictive models that capture the temporal dependencies between students' dynamically changing problem-solving behaviors. We find that the strategy-based temporal analytics framework outperforms baseline models that do not capture strategic temporal dependencies on predictive accuracy. Further, we find that the strategy-based temporal analytics framework utilizing both student problem-solving behavior traces and pre-test performance outperforms a model that uses only pre-test data. The results suggest that strategy-based temporal analytics can serve as the foundation for effective stealth assessment in game-based learning.

2. RELATED WORK

Game-based learning leverages game design elements to foster engagement in learning [7]. Because of its potential to create motivating learning experiences, game-based learning has been explored for a broad range of subjects including science [29], mathematics [17], computer science [3, 24], and public policy [36]. A notable family of game-based learning environments, intelligent game-based learning environments, integrate intelligent tutoring system functionalities and game-based learning [15, 20]. Intelligent game-based learning environments can embed stealth assessments, which have emerged as a promising approach to assessing game-based learning [37, 31, 39]. In stealth assessment, student competencies are assessed unobtrusively by drawing inferences from observations of students' learning interactions.

In one approach to stealth assessment, a directed graphical model was built based on relevant competencies, and related variables were extracted from the observed data to be used as evidence for the targeted competencies [19]. In another approach, Falakmasir and colleagues investigated two hidden Markov models (HMMs) that were trained for high-performing and low-performing students [12]. Subsequently, for observed sequences of events, log-likelihoods were calculated for each HMM. Finally, the difference between the two log-likelihoods was used in a linear regression model to predict post-test scores. This approach reduces the need for labor-intensive domain knowledge engineering.

Work on deep learning-based stealth assessment, DeepStealth, offers an alternate approach that uses artificial neural networks to perform stealth assessment [24]. DeepStealth used a deep feedforward neural network (FFNN) to learn multi-level, hierarchical representations of the input data for evidence modeling. In subsequent work, structural limitations in the FFNNs were addressed with a long short-term memory network-based stealth assessment framework that directly uses students' raw interaction data as input [25]. The strategy-based temporal analytics framework we propose in this paper builds on this prior work, but while the previous work focused primarily on computational methods to model evidence within ECD, the approach introduced in this paper derives temporal evidence from students' dynamic in-game strategy use throughout their problem solving. We cluster students to categorize them based on in-game strategy utilization per task, and then use sequences of in-game strategy features over multiple tasks to predict post-test performance.

Previous work has also explored approaches to detect students' problem-solving strategies using trace data. For example, one effort focused on building a probabilistic model that jointly represent students' knowledge and strategies [16], which was effective at predicting learning outcomes. Another approach focused on selecting features for classifying students' efficiency in solving challenges [22]. The temporal analytics framework we introduce in the paper uses problem-solving strategies that are automatically discovered through clustering based on *n*-grams of players' sequences of interactions with a game-based learning environment, thus obviating the need for labeling or expert knowledge.

3. EXPERIMENTAL SETUP

We investigate the strategy-based temporal analytics approach for stealth assessment with data collected from middle school students' interactions with a game-based learning environment for computational thinking. We describe the learning environment, its in-game problem-solving challenges, and the dataset generated from students' interactions with the game-based learning environment.

3.1 ENGAGE Game-based Learning Environment

ENGAGE is a game-based learning environment designed to introduce computational thinking to middle school students (ages 11-13) (Figure 1). The game was developed with the Unity multi-platform game engine and features a rich, immersive 3D storyworld for learning computing concepts [3, 24]. The game-based learning environment aims to promote computational thinking skills including abstraction and algorithmic thinking through problem solving and programming. The computational challenges within the game were designed to prepare middle school students for computer science work in high school, and to promote positive attitudes toward computer science.

A diverse set of over 300 middle school students participated in focus group activities, pilot tests, and classroom studies with the game. Of the students who provided demographic information, 47% were female; 24% were African American or Black,16% were Hispanic or Latino/a, 17% were Asian, 38% were White, and 5% of the students were Multiracial. The research team worked closely with a similarly diverse group of teachers throughout the project. A subset of teachers helped to co-design the game-based learning activities, providing iterative feedback throughout development. Each of the teachers implementing the game in their classrooms attended either one or two summer professional development workshops that introduced computational thinking concepts and the ENGAGE game-based learning environment.

In the game, students play the role of the protagonist who is sent to investigate an underwater research facility that has lost communications with the outside world. As students progress through the game, they discover that a nefarious villain has taken control of the computing devices within the facility. Students navigate through a series of interconnected rooms and solve a set of computational challenges. Each of the challenges can be solved either by programming devices or interacting with devices in reference to their pre-written programs. Students use a visual block-based programming language to program the devices [25]. They are supported throughout the game by a cast of non-player characters who help them progress through the narrative, offer clues, and provide feedback while they navigate the game and solve computational challenges [24].

The game consists of three major levels: the Introductory Level, in which students learn the basics of the game and simple programming; the Digital World Level, in which students learn

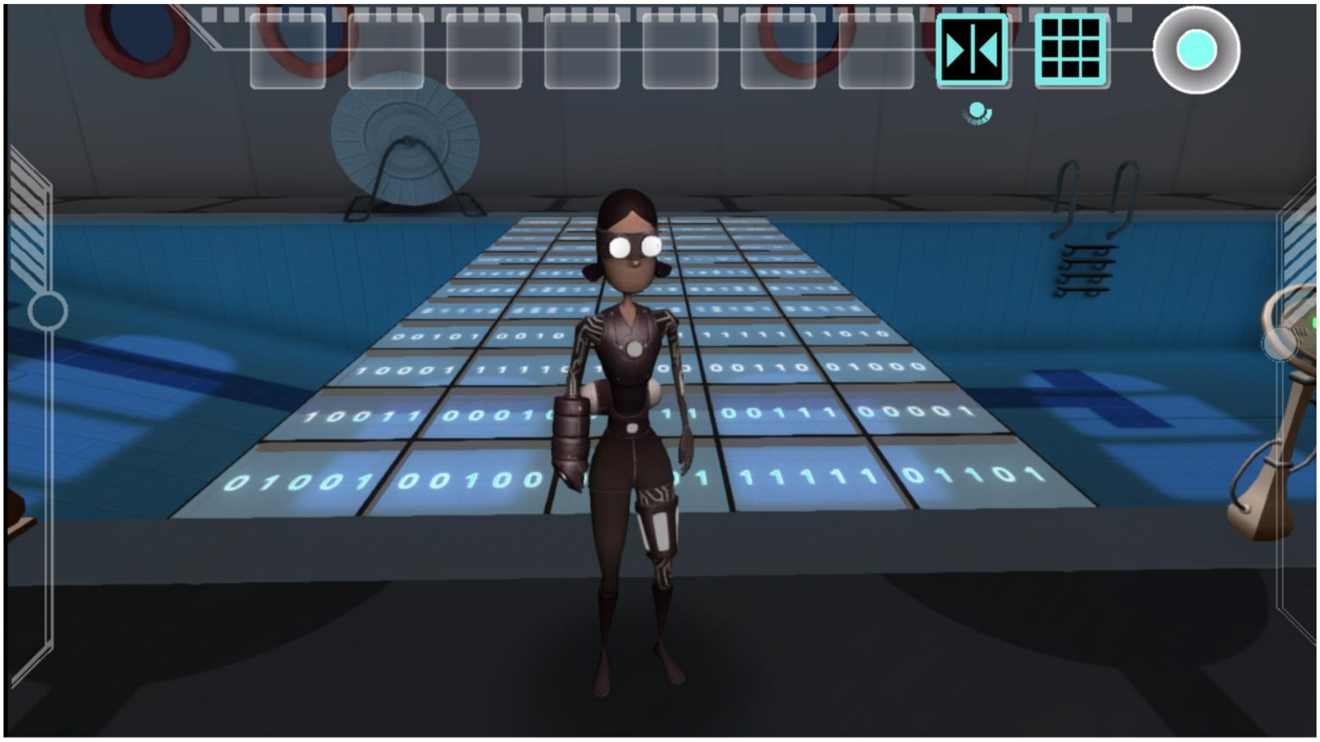


Figure 1. ENGAGE game-based learning environment: students write a program that loops over a binary grid.

how digital data is represented with binary sequences; and the Big Data Level, in which students have the opportunity to work with various datasets and retrieve hidden information by cycling through data and filtering it based on different conditions.

The work presented in this paper focuses on students' problem-solving activities within the Digital World level. The first set of tasks in this level consists of binary locks that are programmed to open if the binary representation of a specific base-ten number is generated by the students. Similarly, the second set of tasks (Figure 2) in the Digital World level features lift devices that are activated when students generate the target base-ten value by flipping binary tiles and execute the program associated with the lift. For example, students can find the target number by reviewing an existing program (Figure 2, right) associated with the binary lift device. Each lift provides students with five consecutive flip tiles representing bits of a five-digit binary number. Players can toggle each bit between 0 and 1 by flipping the corresponding tile on the tile panel. The decimal representation of their generated binary number will be presented on a small screen above the panel. To teach variations of binary representations, the game enables students to flip tiles between '0' and '1', 'black' and 'white', and 'F' (False) and 'T' (True), as in (Figure 2, left).

To advance to the next task, students must flip binary tiles on the binary lift device to generate the target decimal number (Figure 2, left) execute its program, and lift up the binary device. Through these tasks, students learn about the concept of bits in binary numbers and the weight assigned to each bit. In the analyses reported here, we used behavior trace data from students' interactions with 11 binary tasks from the Digital World level, where students learn the weight associated with each of the five bits through the first five tasks and then learn how to combine multiple bits to make more complex numbers with binary representations.

3.2 Dataset

We analyzed 244 students' behavior trace data obtained from a teacher-led study in four public middle school classrooms in the urban area in the United States. The four schools reported an average percentage of free or reduced lunch as 34.75%, 41.07%, 31.65%, and 63.17% during the years of data collection, respectively. Furthermore, three of the schools were magnets for gifted and talented students and the fourth was a magnet for leadership and innovation. To support collaborative learning, which is prominent in computer science education [3], we collected student behavior trace interaction data from pairs of students in which they took turns serving as navigator (traversing the game) and driver (action planning). Pre- and post-test assessments measuring content knowledge (e.g., binary representation) were completed individually by students before starting the Digital World level (pre-test) and immediately after finishing it (post-test). Both pre-test and post-test are on a scale of 0 to 1. Out of 244 students, 168 students completed the pre-test and post-test for content knowledge as well as all 11 binary representation tasks for this level. The results of conducting a paired t-test on students' content knowledge pre-test (M=0.44, SD=0.20) and post-test (M=0.59, SD=0.24) revealed a significant improvement from pre-test to post-test scores ($t(167)$ =11.24, $p<0.001$).

4. MODELING STUDENTS' PROBLEM-SOLVING STRATEGIES

Students exhibited a broad spectrum of problem-solving strategies while solving the binary challenges in the Digital World level. For example, some students pursued random trial-and-error strategies to find solutions, while at the other end of the spectrum, some students pursued thoughtful systematic approaches to solve the challenge. As would be expected, some students fell in the middle of this spectrum by utilizing more thoughtful trial-and-error.

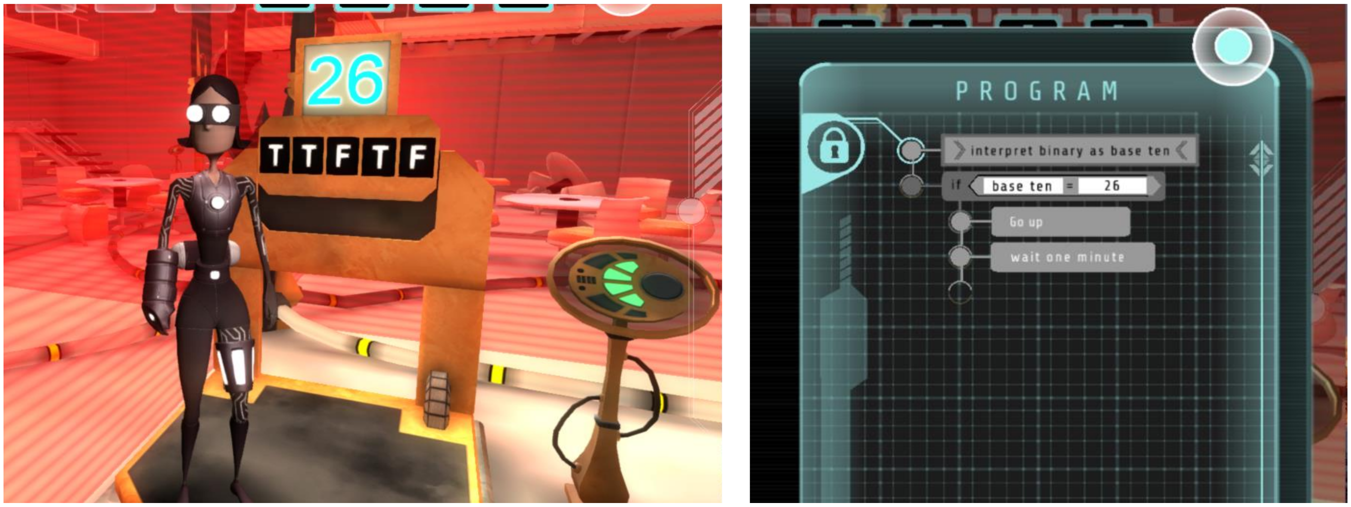


Figure 2. (Left) A binary lock device that students must unlock. The T (true) tiles indicate the bits are 1, whereas F (false) tiles denote 0. The current binary number is 11010 and the corresponding base-ten number, 26, is displayed on the device as immediate feedback. (Right) The visual programming interface displaying the binary lock's program.

For each of the 11 consecutive binary challenges, we used students' tile-flip sequences to cluster them into distinct groups. Subsequently, we interpreted these clusters in terms of the problem-solving strategy exhibited by members of each cluster. Below we first describe the process of clustering students' task-level strategies based on their binary tile flip sequences and then describe the representative problem-solving strategy in each cluster.

4.1 Methodology

In order to group students based on their problem-solving strategies, we first derived features from students' binary tile flip sequences. We encoded the flip sequences as *n*-grams, commonly used as a representation for sequential data such as text and speech [41], as well as for sequential trace data [12]. The *n*-gram representation extracts sequences of *n* adjacent elements from the original string. We consider each unique *n*-gram as a feature in our *n*-gram based feature vector, while we use the frequency of each *n*-gram occurrence in a flip string as a value in this work.

For each of the 11 binary challenges, students' behaviors (i.e., the flip sequence generated for that specific task) were clustered based on the extracted *n*-gram features, resulting in 11 sequential cluster-memberships per student. Since each task differed slightly from the other tasks, we analyzed students problem-solving behavior separately for each task. In the following sections, we describe how we identified different problem-solving strategies using the proposed clustering method.

4.1.1 Feature Engineering

We extracted students' interactions with binary flips in the format of a string containing students' consecutive flips of the binary tiles for each task. Each task is associated with a decimal number to operate the device (e.g., 26 in Figure 2), where the binary number displayed on the five tiles is set to 00000 by default. For example, considering tiles' indices starting at one from the right most tile, if a student has flipped tile number four (i.e., 01000 with the decimal representation of 8), followed by flipping tile number five (i.e., 11000 with the decimal representation of 24), their tile flip string would become {4, 5}.

In order to capture the most fine-grained information present in the series of flips, we used *n*-grams with varying lengths of *n*. Preliminary explorations showed including sequences of lengths larger than four exponentially increases the sparsity of the dataset. To eliminate the sparsity issue, we capped the *n*-gram size at 4. Our final feature set ranges from sequences of length one (i.e., unigram features) to sequences of length four (i.e., 4-grams) that are repeated at least three times throughout our dataset. We used the natural language processing toolkit (NLTK) library for Python to extract *n*-grams and their associated frequency from each flip string. For example, for one of the tasks, a total of 2,495 unique *n*-grams with at least three occurrences were generated from the student flip strings for that task. These *n*-gram feature vectors were then used to cluster students' in-game strategy use per task, where an *n*-gram feature vector per student was generated separately for each of the 11 tasks.

Flip strings provide a fine-grained representation of students' problem-solving behaviors in solving binary representation challenges, and these features offer a method to identify students' adopted strategies. As an example, consecutive flips of the same tile by a student can be an indicator of the student's intention to learn the weight assigned to that binary digit. Further, the overall number of flips conducted to generate the target base-ten value can be used to gauge the students' overall efficiency in solving the problem.

4.1.2 Clustering

Next, we applied the expectation-maximization (EM) clustering technique to students' flip behaviors represented using an *n*-gram feature vectors to identify students' problem-solving strategies. Because each of the 11 tasks in the Digital World level targets a different base-ten number, the binary code needed to solve the task is different. Consequently, flip sequences obtained from students' interactions with a binary device reveal information specific to the target value designed for the task. Thus, clustering was performed separately for each of the 11 tasks. We used the MClust package in R to cluster the feature vectors. EM clustering can explore a range of cluster numbers and return the (local) optimal number of clusters based on the maximum likelihood estimation. A different optimal number of clusters was identified for each task. Three, four, and nine clusters emerged most frequently when we explored the number of clusters between two to ten. A preliminary investigation on these different number of clusters found that three clusters

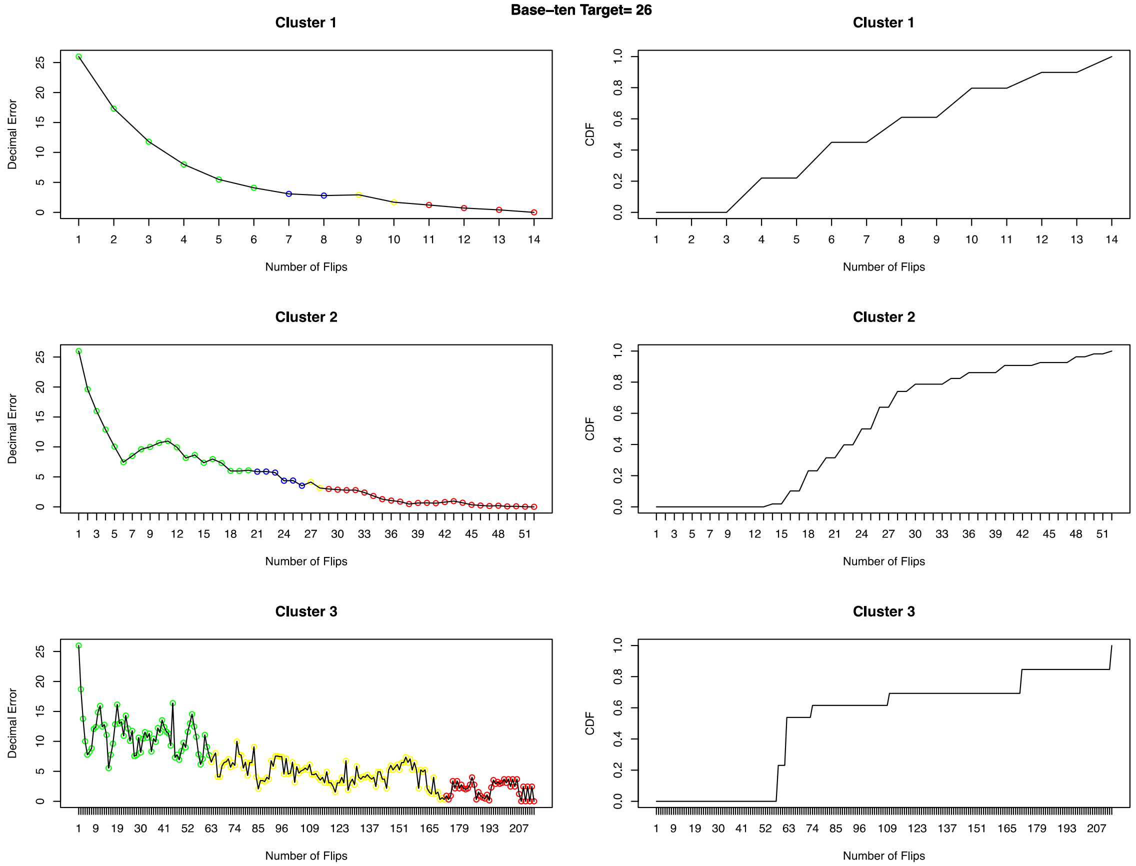


Figure 3. Students' average decimal error and the CDF of the present population at each flip.

showed coherent patterns for problem-solving strategies across all tasks, and we thus chose three as the number of clusters for all tasks in this work.

4.2 Interpreting Clusters

To interpret student problem-solving strategies using the clusters as identified above, we present two novel methods that measure the error between the target value and the student-generated value. To analyze students' problem-solving patterns for each cluster of each task, we calculated the average error at each flip relative to the solution target of students who belong to the same cluster. All students start from the difference between the default value zero and the target decimal value. Since we analyzed students who completed a task, the average error for each cluster decays toward zero, and we expected to observe distinct error-decay patterns across the three clusters. We introduce two error calculation metrics that measure students' error based on the distance from the current value to the targeted value: the decimal error and binary error. These two approaches are described below.

4.2.1 Decimal Error

The decimal error is the absolute difference between the target base-ten value and the base-ten representation of the student-generated binary string. Each student starts with an error equal to the target value and ends with an error equal to zero. We calculate the decimal error after every new flip. As a result, a sequence of decimal errors is generated for each student per flip action when completing each of the 11 tasks. We then plot the average decimal error where the y-axis shows students in the same cluster (separately for each task), and the x-axis shows the maximum number of flips observed in the cluster as in Figure 3 (left). Because the total number of flips is different for each student in a cluster, we use the decimal error value of zero for students who already completed the task and calculate the average decimal error over all students in the cluster.

For example, suppose there are two students in a cluster, where student A's decimal error sequence is {2, 1, 2, 0} and student B's error sequence is {2, 0} in the task of making the value two in base-ten. We use the maximum length of sequence, four, obtained from student A, and reformulate student B's sequence to {2, 0, 0, 0}. In this case, the average decimal error sequence becomes {2, 0.5, 1, 0}. The average error at each flip for the eighth binary challenge where the student is asked to find the binary number for the target 26 is shown in Figure 3 (left). For this task 118 students were grouped in the first cluster, 108 students were grouped in the second cluster, and 19 students were grouped in the third cluster. In Figure 3, because the target value for this task is 26, the average error for students is 26 in the beginning, which becomes zero at the end, while decay patterns differ across clusters. Students in each cluster solved the problem within a varying number of flips. The error for students who finished earlier is represented with zero. We show the percentage of students still working on the challenge at each flip using a color coding scheme. In Figure 3, green points mark flips where between 70% to 100% of the population is present, blue points indicate the presence of 50% to 70% of the population, yellow points mark 30% to 50% of the population, and red points indicate flips were less than 30% of the population of that cluster are still working on the problem. These percentages are derived from the cumulative density functions (CDFs) of clusters'

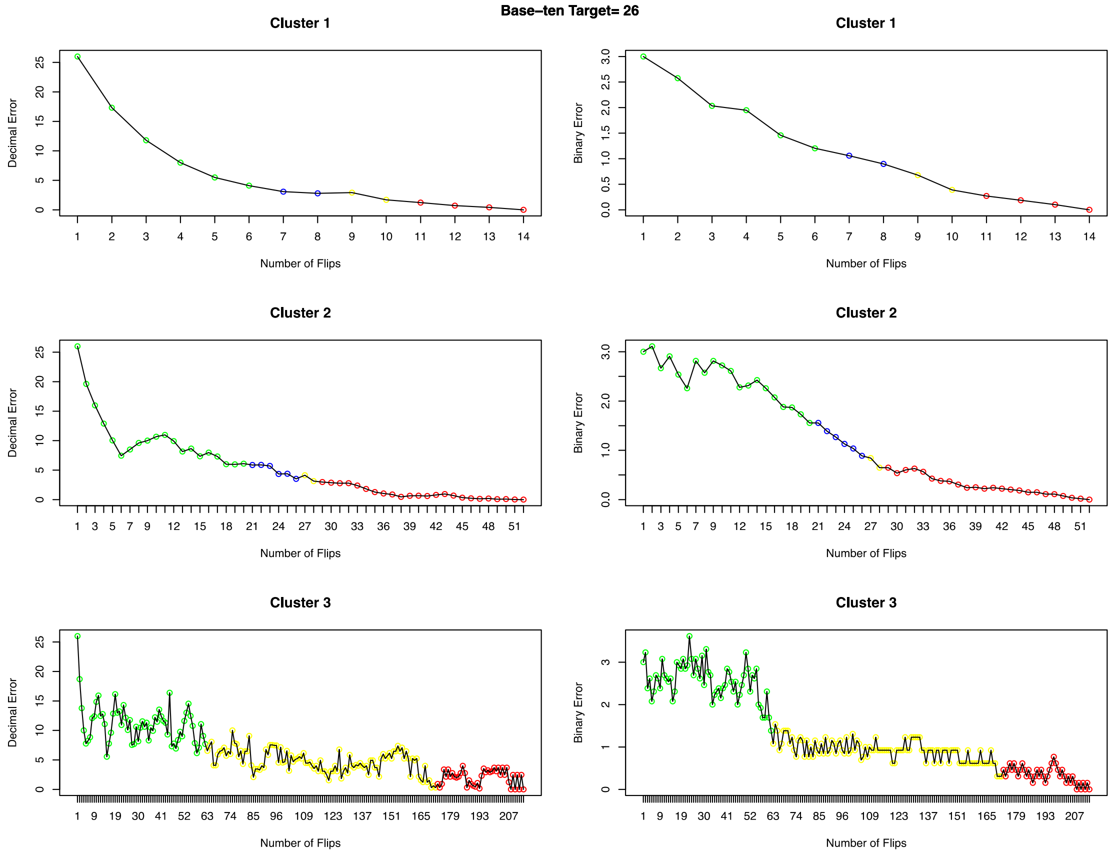


Figure 4. Students' average binary and decimal error at each flip.

present population at each flip that are plotted in Figure 3 (right). The clusters for the other ten binary representation tasks in the Digital World level follow similar error-decay patterns.

4.2.2 Binary Error

Binary error is the Hamming distance, the number of different elements in two strings with the same size, between the current state of the student-generated binary string and the binary representation of the target base-ten value. The approach for plotting the binary error is similar to the approach for plotting the decimal error. Figure 4 shows binary (right) and decimal (left) error plots for each cluster of the challenge, finding the binary representation of the number, 26. As the binary error plots are generated from the same cluster-based population used for generating decimal error plots, the same CDFs as in (Figure 3, right) hold for binary error plots.

4.2.3 Resulting Strategies

The same general patterns apply to other challenges analyzed in this study. As seen in Figure 4, there is a coherence in the error decay pattern between the decimal and the binary error. The decimal error captures students' strategies to make the base-ten errors between the target number and current binary representations as small as possible, while the binary error places more emphasis on the representational difference between binary sequences focusing on students' understanding on each bit and its associated weight. The analyses reveal a clear distinction in students' problem-solving strategies in solving the in-game challenges. After clustering, the following distinct groups emerge for all analyzed tasks:

- Students who completed the task more quickly than the other groups and with fewer trial-and-error attempts (Cluster 1).
- Students who had a moderate number of flips and demonstrated error decaying continuously toward zero with some trial-and-error attempts (Cluster 2).
- Students who completed the activity with many more flips compared to students in the other clusters, which may be an indicator of less thoughtful trial-and-error attempts (Cluster 3).

The binary and the decimal error decay patterns paralleled each other for every cluster of every task. The analyses reveal that the two error metrics similarly capture *n*-gram-encoded student behaviors, while students' per-task behaviors naturally fall into one of the three groups. We used these identified clusters as game strategy features for our evidence model for stealth assessment.

5. STEALTH ASSESSMENT

Modeling students' strategies can contribute to improving their learning outcomes [11, 33]. In this work, we aim to evaluate the predictive power of models of students' in-game problem-solving strategies over time to predict their post-test performance. We seek to determine if the in-game strategies observed in students' interactions with the game environment can be used as evidence for stealth assessment.

The feature set implicitly represents rich temporal dependencies among student behaviors over the course of interactions with the

ENGAGE game-based learning environment. To effectively model temporal dependencies in the feature set we investigate an evidence model based on long short-term memory network (LSTM) [14] to infer students' post-test performance based on their in-game strategy use over time. We also examine two baseline classification techniques, random forest (RF) [1] and support vector machines (SVM) [9], to predict students' post-test performance. It is important to note that, in contrast to the LSTM approach, neither the random forest nor the support vector machine approaches explicitly capture the temporal relationships in students' strategies. Thus, they treat in-game strategy features as independent features in their predictions.

We devise two LSTM-based evidence models, one model utilizing a feature set that contains pre-test performance only and another model utilizing both pre-test performance and in-game strategy features, to isolate the effects of incorporating the temporal dependencies captured by the LSTM-based model.

5.1 Data Preparation

For the classification task, we use data from 168 students who finished both pre- and post-tests and also completed all 11 binary challenge tasks within the game. We excluded data for students who did not complete all 11 tasks as we intended to perform a temporal analysis across these tasks. Initially, the dataset included pre- and post-test scores along with students' flip sequences, which were then transformed into *n*-gram features for each of the 11 tasks. We divide the data into training and held-out test sets. We first perform clustering using students' *n*-gram feature vectors in the training set. After identifying distinct clusters for each challenge in the training set, we use the Gaussian finite mixture models estimated by the MClust package to cluster students' data in the test set. This maintains the independence of the training and test sets. Students' data in both the training and test sets are represented with sequences of in-game strategies across the 11 binary-representation tasks along with their pre-test performance (i.e., high, medium, low), a categorical representation of the pre-test score based on a tertile split obtained from the distributions of the pre-test scores. We use these input features to predict post-test performance also using the three labels, which are obtained based on a tertile split of students' post-test scores. We chose tertiles to create a balanced distribution among all classes. The initial pre- and post-test scores are continuous variables, ranging between 0 to 1. For the pre-test, scores between ($0 \leq$ score ≤ 0.36) are categorized as low, scores between ($0.36 <$ score ≤ 0.54) as medium, and scores between ($0.54 <$ score ≤ 1.00) as high. Similarly, for the post-test, scores between ($0 \leq$ score ≤ 0.45) are categorized as low, scores between ($0.45 <$ score ≤ 0.72) as medium, and scores between ($0.72 <$ score ≤ 1.00) are categorized as high. Table 1 presents the distribution of students (*n*=168) with respect to students' pre- and post-test performance.

Table 1. Distribution of students (*n*=168) in relation to their pre- and post-test performance

Test	Low	Medium	High
Pre-test	74	53	41
Post-test	60	59	49

To transform the data into a trainable representation, we use one-hot encoding on the categorical variables (i.e., pre-test performance and the 11 in-game strategy changes) in preparation for the classification task. One-hot encoding is a feature representation method for a categorical variable, where a feature vector whose length is the size of the possible values is created, and only the associated feature bit is on (i.e., 1) while all other feature bits are off (i.e., 0). We also prepare two distinct feature sets to evaluate the predictive power of the in-game strategy features:

- Full feature set: For RF and SVM, 36 features including 33 one-hot encoded features representing the cluster membership among the three clusters for each of the 11 binary tasks and three one-hot encoding-based features (i.e., low, medium, and high) representing students' pre-test performance and 3 features to represent students' pre-test performance. For LSTMs, since they take as input the pre-test performance (3 features) and a task-specific in-game strategy (3 features) per time step, it utilizes six features.
- Pre-test performance feature set: Three one-hot encoding-based features (i.e., low, medium, and high) representing students' pre-test performance.

5.2 Classification Methods

We use 'randomForest' [21] and 'e1071' [23] packages in R to train random forest and SVM classifiers, respectively. For LSTM-based evidence models, we use the Keras [6] and scikit-learn [30] libraries in Python.

We use 5-fold cross-validation within the training data to tune the hyperparameters of the classification techniques based on the full feature set. After optimizing the hyperparameters, we train each of the classifiers using the full training set and evaluate them on the held-out test set. After comparing classifiers, we take the best performing classification technique and train an additional model based on the other feature set, pre-test performance feature-set, using the same test/train data split used for the full feature set-level analysis. The classification process for each classifier and their results are described below.

5.2.1 Baseline Method

The majority class-based method assigns the most frequent label in the dataset as the predicted label for all data instances. Since the most common label is the grade 'low', all labels will simply be predicted as the first class (i.e., low post-performance). The result of applying the baseline method on the full feature set achieves an accuracy of 35.71%. The macro average for recall is 33.33%. The precision and F1-score are undefined here since the baseline method predicts the most frequent label for all instances, while producing no other labels.

5.2.2 Random Forest Method

The random forest technique generates multiple decision trees using different subsets of the training data using bagging. A random forest tree is generated by trying a random subset of available features at each split. It then classifies each point in the test set using all the trees and uses the majority vote for classifying the test point. We use the set (10, 25, 50, 100, 200) to tune the number of trees for the model. Using a 5-fold cross-validation approach on our training set we found 25 to be the best number of trees for the full feature set.

Random forest classifiers are subject to randomness when being trained on a dataset. They perform feature bagging (i.e., a random selection of the features at each candidate split), and thus the predictive performance of random forests trained utilizing the same set of hyperparameters can vary depending on the random procedure. As a result, each round of training and evaluation on the same training/test sets will result in slightly different accuracies. Hence, we report the average result of 100 rounds of training and evaluating the classifier. The mean and standard deviation of the results are shown in Table 2. The results of applying the random forest classifier on the full feature set achieve an average accuracy of 50.43%, an average precision of 52.20%, an average recall of 50.98%, and an average F1-score of 51.03%. The precision, recall and F1 measures are calculated using a macro-average of all three classes (i.e., simple average of the relative measurement of all three classes).

5.2.3 SVM Method

Support vector machines (SVMs) can be used for both regression and classification tasks. In classification tasks for which data are not linearly separable, data will be transformed to a higher-dimensional space for linear separability, and SVMs are applied to classify the transformed data. For this classification task, we use a third-degree polynomial kernel. We tune *C* as the hyperparameter of our SVM model. *C* is the regularization parameter that controls models' tolerance for incorrect classifications during training. We explore a set of values (0.01, 0.1, 1, 10, 15) to tune *C* on the full feature set. Using a 5-fold cross-validation approach on the training set, we found *C* = 1 to be the best parameter to be used in the model. The results of applying the SVM model on the test set show an accuracy of 41.17%, a precision of 44.14%, a recall of 39.25%, and an F1-score of 35.44%. Similar to the RF classifier we report the average result of 100 rounds of training and evaluating the trained classifier. Since there is no random parameter for this method, the standard deviation for the estimated accuracies is 0.

Both the random forest and SVM approaches achieve higher accuracies compared to the simple majority class baseline, suggesting that these methods are effective for stealth assessment. We hypothesize that the accuracy could be increased by explicitly modeling the temporal relationships across students' sequential problem-solving tasks. We next describe the LSTM-based approach and the results it produces.

5.2.4 LSTM Method

LSTMs are a type of recurrent neural networks (RNNs), a class of deep learning methods that are capable of learning temporal patterns in data. This characteristic makes LSTMs a promising candidate for classifying sequential data, such as time-series data of students' strategy uses across the 11 binary challenges they solve during gameplay. A sequence of cluster types (i.e., in-game problem-solving strategies for the 11 in-game binary representation tasks) can reveal students' problem-solving progressions as they unfold over learning sessions to predict students' learning outcomes. We investigate LSTMs to model dynamic changes in students' problem-solving strategies, motivated by LSTMs' ability to preserve long-term dependencies through their three gating units (i.e., input, forget, and output gates).

We tune the number of LSTM layers and the number of hidden units within each layer by conducting a 5-fold cross validation on the training set. We explore 15 different hyperparameter combinations with different numbers of hidden layers (1, 2, 3) and different numbers of hidden units in each layer (10, 15, 25, 50, 100). We found that networks with 2 layers with 15 units per produced the best results for predictive accuracy.

Like random forest models, the LSTM-based approach also results in different models each time it is trained on the same training set. Hence, evaluating these models on the same test-set generates slightly different outputs. This is due to the fact that deep learning approaches are sensitive to the random weights used to initialize the network. In addition, these types of techniques are trained on batches and the input order of the batches influence the models that are generated. We report an average of 100 runs of training and evaluating the LSTM classifier on the same training and test set. The results of applying this LSTM on the held-out test set achieve an average accuracy of 64.82%, an average precision of 63.88%, an average recall of 65.14%, and an F1-score of 63.68%.

Table 2 provides a summary of the results of the classification methods. The highest score per metric is indicated in bold. The baseline and SVM approaches are deterministic so their metrics' standard deviations are zero. All classification methods outperform the majority class baseline. Because reasoning about students' problem-solving strategy adoption *over time* can inform predictions about the strength of their learning as measured by post-test performance, the LSTM-based evidence model yields considerable improvement over the other approaches. The results indicate that the LSTM model appears to successfully capture the latent temporal dependencies among features in students' problem solving.

Table 2. Performance (± standard deviation) of classifiers

Method	Accuracy	Precision	Recall	F1
Baseline	35.7(±0.0)	N/A	33.3(±0.0)	N/A
RF	50.4(±2.5)	52.2(±2.7)	51.0(±2.4)	51.0(±2.6)
SVM	41.2(±0.0)	44.1(±0.0)	39.3(±0.0)	35.4(±0.0)
LSTM	**64.8**(±2.7)	**63.9**(±2.8)	**65.1**(±2.8)	**63.7**(±2.5)

5.3 In-game Strategy for Stealth Assessment

To further investigate the effectiveness of the in-game strategy features in predicting students' post-test performance, we compare two versions of the LSTM-based model, our best performing classification technique. We create a version of the LSTM-based model trained on the full feature set (pre-test features together with in-game strategy features) and compare it to a partial feature set version (pre-test features only). The results of this evaluation are shown in Table 3, where the highest score per metric is indicated in bold.

Table 3. Results of applying LSTM on pre-test only, in-game strategy, and full features feature sets

Feature set	Accuracy	Precision	Recall	F1
Full FS	**64.8** (±2.7)	**63.9**(±2.8)	**65.1**(±2.8)	**63.7**(±2.5)
Pre-test FS	44.7(±7.9)	N/A	42.8(±8.3)	N/A

The results demonstrate that incorporating the in-game strategy features into the model significantly contributes to predictive accuracy. Compared to the 44.66% accuracy achieved by the partial feature set version (pre-test features only), the model that uses in-game strategy features in addition to pre-test features achieves an accuracy of 64.82%. The significantly higher accuracy achieved by the full-set model suggests that the strategy-based approach that uses sequences of strategies as represented by strategy clusters appears to capture an important quality of students' problem-solving strategies that are predictive of learning performance.

6. DISCUSSION

Stealth assessment relies on accurate evidence models inferred from student behavior traces, and we found that student behavior traces can serve as the foundation for evidence models that are driven by students' in-game problem-solving strategies.

Building on previous work on stealth assessment we have presented a novel problem-solving-strategy-based temporal analytics framework leveraging a clustering approach, which notably does not require a labor-intensive process of labeling data. While the previous work focused on computational methods to model evidence within ECD using deep learning networks, we have investigated temporal evidence derived from students' dynamic in-game strategy uses throughout their game play, and have demonstrated the effectiveness of LSTM-based evidence models that predict students' post-test performance.

For each of the 11 problem-solving tasks in the ENGAGE game-based learning environment, we first transformed sequences of student behavior interactions into sequences of *n*-gram features to capture the temporal information that spans interaction sequences and clustered them with EM Clustering. The results revealed clear distinctions in students' approaches toward solving these computational thinking challenges. The clustering grouped students into those who solved the problem in a few flips and a few attempts, those who solved the problem with a moderate number of flips and with thoughtful trial-and-error, and those who solved the problem with a long sequence of flips and with seemingly random trial-and-error. While in our game settings students could try the problems as many times as they wanted, other game environments might take number of trials into account using a point system that could affect players' problem-solving strategies.

We then used students' cluster memberships across different tasks as an indicator of their in-game problem-solving strategy and used these problem-solving strategies to inform the evidence model for predicting students' post-test performance. The results demonstrated that the in-game strategy features provide strong predictive capacity for LSTM-based evidence models and more generally for the use of stealth assessment. It has been shown that LSTM-based ECD evidence models with in-game strategy features effectively capture the temporal relationships between strategies, as supported by the models' highest predictive accuracy rate, precision rate, recall rate, and F1 scores outperforming competitive non-sequential baseline approaches in predicting students' post-test performance. We used a relatively small dataset, 168 students for this analysis. after collecting more data, we can further verify our results.

It is important to note that the in-game strategy features are derived directly from log data and are generated based on an unsupervised method, EM Clustering. This automated process of extracting students' in-game problem-solving strategy makes it a promising approach for evidence modeling. The approach can be readily used for evidence modeling design for learning environments that center on students solving problems by performing sequences of actions from a limited pool of available actions. However, the proposed approach is not appropriate for analyzing ill-defined problems where players are not bound to certain actions.

Evidence models such as those induced in this paper can be used by intelligent game-based learning environments to infer students' problem-solving strategies from trace data analysis. When the learning environments are signaled by the evidence models that a student is following a strategy associated with a poor learning outcome, it can intervene to guide students towards more productive strategies. In addition to strategy scaffolding, the evidence models can also work in tandem with knowledge modeling to support knowledge scaffolding. For example, in the ENGAGE game-based learning environment, students' generating a desired binary sequence through long series of flips and random trial-and-error might be an indicator of lack of knowledge about digit weights in a binary string, which could be addressed with a timely explanation of binary digit weights. The results of the work reported here, as well as those found in related work on inferring student problem-solving strategies from behavior trace data [18], suggest that modeling students' problem-solving strategies may contribute to improved assessment and also lead to learning environments that can adapt more effectively to students' needs.

7. CONCLUSION

Stealth assessment holds considerable potential for game-based learning. Although high volumes of dynamic student interaction data can be readily captured from game-based learning environments, effective stealth assessment poses significant challenges. We have introduced a strategy-based temporal analytics framework for stealth assessment that uses an LSTM-based evidence model trained on sequences of student problem-solving strategies learned from clustering *n*-gram representations of student in-game behaviors. In an evaluation of predictive accuracy for student learning, the strategy-based temporal analytics framework outperformed baseline models that did not capture the temporal dependencies of strategy use. In future work, it will be important to investigate multiple granularities of strategy representations that may lend themselves to hierarchical deep learning methods. It will also be instructive to incorporate the LSTM-based models into game-based learning environments to explore how they can provide classic stealth assessment functionalities while simultaneously supporting adaptive scaffolding.

8. ACKNOWLEDGMENTS

This research was supported by the National Science Foundation under Grants CNS-1138497 and DRL-1640141. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

9. REFERENCES

[1] L. Breiman. Random forests. *Machine Learning*, *45*(1):5-32, 2001.

[2] P. Brusilovsky and E. Millán. User models for adaptive hypermedia and adaptive educational systems. *The Adaptive Web*, pages 3–53, 2007.

[3] P. Buffum, M. Frankosky, K. Boyer, E. Wiebe, B. Mott, and J. Lester. Collaboration and gender equity in game-based learning for middle school computer science. *IEEE Computing in Science and Engineering*, 18(2), 18-28, 2016.

[4] G. Chen, S. Gully, and D. Eden. Validation of a new general self-efficacy scale. *Organizational Research Methods*, 4(1):62–83, 2001

[5] M. Cheng, L. Rosenheck, C. Lin, and E. Klopfer. Analyzing gameplay data to inform feedback loops in the radix endeavor. *Computers & Education*, 111:60–73, 2017.

[6] F. Chollet. Keras. https://github.com/keras-team/keras, 2015

[7] D. Clark, E. Tanner-Smith, and S. Killingsworth. Digital games, design, and learning: A systematic review and meta-analysis. *Review of Educational Research*, 86(1):79–122, 2016.

[8] D. Cordova and M. Lepper. Intrinsic motivation and the process of learning: Beneficial effects of contextualization,

personalization, and choice. *Journal of Educational Psychology*, 88(4):715–730, 1996.

[9] C. Cortes and V. Vapnik, Support-vector networks. *Machine Learning*, *20*(3):273-297, 1995.

[10] M. d'Aquin and N. Jay. Interpreting data mining results with linked data for learning analytics: motivation, case study and directions. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, pages 155–164, 2013.

[11] M. Eagle and T. Barnes. Exploring differences in problem solving with data-driven approach maps. In *Proceedings of the 7th International Conference on Educational Data Mining*, pages 76–83, 2014.

[12] M. Falakmasir, J. Gonzalez-Brenes, G. Gordon, and K. DiCerbo. A data-driven approach for inferring student proficiency from game activity logs. In *Proceedings of the Third ACM Conference on Learning@ Scale*, pages 341–349. 2016.

[13] J. Harley, C. Carter, N. Papaionnou, F. Bouchet, R. Landis, R. Azevedo, and L. Karabachian. Examining the predictive relationship between personality and emotion traits and learners' agent-direct emotions. In *proceedings of the 7th International Conference on Artificial Intelligence in Education*, pages 145–154, 2015.

[14] S. Hochreiter, J. Schmidhuber. Long short-term memory. *Neural Computation*, *9*(8):1735–1780, 1997.

[15] G. Jackson and D. McNamara. Motivation and performance in a game-based intelligent tutoring system. *Journal of Educational Psychology*, 105(4):1036–1049, 2013.

[16] T. Käser, N. Hallinen, and D. Schwartz. Modeling exploration strategies to predict student performance within a learning environment and beyond. In *Proceedings of the 7th International Conference on Learning Analytics and Knowledge*, pages 31–40, 2017.

[17] M. Kebritchi, A. Hirumi, and H. Bai. The effects of modern mathematics computer games on mathematics achievement and class motivation. *Computers & Education*, 55(2):427–443, 2010.

[18] D. Kerr and G. Chung. Identifying key features of student performance in educational video games and simulations through cluster analysis. *Journal of Educational Data Mining*, 4(1):144–182, 2012.

[19] Y. Kim, R. Almond, and V. Shute. Applying evidence-centered design for the development of game-based assessments in physics playground. *International Journal of Testing*, 16(2):142–163, 2016

[20] J. Lester, E. Ha, S. Lee, B. Mott, J. Rowe, and J. Sabourin. Serious games get smart: Intelligent game-based learning environments. *Artificial Intelligence Magazine*, *34*(4):31–45, 2013.

[21] A. Liaw and M. Wiener. Classification and regression by randomForest. *R News*. 2(3):18–22, 2002.

[22] L. Malkiewich, R. Baker, V. Shute, S. Kai, and L. Paquette. Classifying behavior to elucidate elegant problem solving in an educational game. In *Proceedings of the 9th International Conference on Educational Data Mining*, pages 448–453, 2016.

[23] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and Friedrich Leisch. e1071: Misc functions of the Department of Statistics, Probability Theory Group, TU Wien, 2017.

[24] W. Min, M. Frankosky, B. Mott, E. Wiebe, K. Boyer, and J. Lester. DeepStealth: Leveraging deep learning models for stealth assessment in game-based learning environments. *In proceedings of the 7th International Conference on Artificial Intelligence in Education*, pages 277-286, 2015.

[25] W. Min, M. Frankosky, B. Mott, E. Wiebe, K. Boyer, and J. Lester. Inducing stealth assessors from game interaction data. In *proceedings of the 9th International Conference on Artificial Intelligence in Education*, pages 212-223, 2017.

[26] W. Min, B. W. Mott, J. P. Rowe, B. Liu, and J. C. Lester. Player goal recognition in open-world digital games with long short-term memory networks. In *proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 2590–2596, 2016.

[27] R. Mislevy, L. Steinberg, and R. Almond. Focus article: on the structure of educational assessments. *Measurement: Interdisciplinary Research and Perspectives*, 1(1):3–62, 2003.

[28] N. Nagappan, L. Williams, M. Ferzli, E. Wiebe, K. Yang, C. Miller, and S. Balik. Improving the CS1 experience with pair programming. *In Proceedings of 34th SIGCSE Technical Symposium*, pages 359–362, 2003.

[29] B. C. Nelson, Y. Kim, C. Foshee, and K. Slack. Visual signaling in virtual world-based assessments: The save science project. *Information Sciences*, 264:32–40, 2014.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, and J. Vanderplas. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 2825-2830, 2011.

[31] E. Quellmalz, M. Timms, M. Silberglitt, and B. Buckley. Science assessments for all: Integrating science simulations into balanced state science assessment systems. *Journal of Research in Science Teaching*, 49(3):363–393, 2012.

[32] D. Quigley, J. Ostwald, and T. Sumner. Scientific modeling: using learning analytics to examine student practices and classroom variation. In *Proceedings of the 7th International Conference on Learning Analytics and Knowledge*, pages 329–338, 2017.

[33] E. Rowe, R. Baker, J. Asbell-Clarke, E. Kasman, and W. Hawkins. Building automated detectors of gameplay strategies to measure implicit science learning. In *Proceedings of the 7th International Conference on Educational Data Mining*, pages 337–338, 2014.

[34] J. Rowe, L. Shores, B. Mott, and J. Lester, Integrating learning, problem solving, and engagement in narrative-centered learning environments. *International Journal of Artificial Intelligence in Education*, *21*(1-2):115-133, 2011.

[35] A. Rupp, M. Gushta, R. Mislevy, and D. Shaffer. Evidence-centered design of epistemic games: Measurement principles for complex learning environments. *The Journal of Technology, Learning and Assessment*, 8(4), 2010.

[36] J. Sabourin and J. Lester. Affect and engagement in game-based learning environments. *IEEE Transactions on Affective Computing*, 5(1):45–56, 2014.

[37] S. Sahebi, Y. Huang, and P. Brusilovsky. Predicting student performance in solving parameterized exercises. In *proceedings of the 12th International Conference on Intelligent Tutoring Systems*, pages 496–503, 2014.

[38] V. Shute. Stealth assessment in computer-based games to support learning. *Computer Games and Instruction*, 55(2):503–524, 2011.

[39] V. Shute and M. Ventura. *Measuring and supporting learning in games: Stealth assessment*. The MIT press, 2013.

[40] P. Wouters, C. Van Nimwegen, H. Van Oostendorp, and E. Van Der Spek. A meta-analysis of the cognitive and motivational effects of serious games. *Journal of educational psychology*, *105*(2), 249–265, 2013.

[41] Z. Xing, J. Pei, and E. Keogh. A brief survey on sequence classification. *ACM SIGKDD Explorations Newsletter*, 12(1): 40–48,2010

Knowledge Tracing Using the Brain

David Halpern* , Shannon Tubridy, Hong Yu Wang,
Camille Gasser, Pamela Osborn Popp, Lila Davachi, Todd M. Gureckis
Department of Psychology
New York University
New York, NY, 10003
david.halpern@nyu.edu, st704@nyu.edu, hyw248@nyu.edu, camille.gasser@nyu.edu, pamop@nyu.edu,
lila.davachi@nyu.edu, todd.gureckis@nyu.edu

ABSTRACT

Knowledge tracing is a popular and successful approach to modeling student learning. In this paper we investigate whether the addition of neuroimaging observations to a knowledge tracing model enables accurate prediction of memory performance in held-out data. We propose a Hidden Markov Model of memory acquisition related to Bayesian Knowledge Tracing and show how continuous functional magnetic resonance imaging (fMRI) signals can be incorporated as observations related to latent knowledge states. We then show, using data collected from a simple second-language learning experiment, that fMRI data acquired during a learning session can be used to improve predictions about student memory at test. The fitted models can also potentially give new insight into the neural mechanisms that contribute to learning and memory.

1. INTRODUCTION

A shared goal for both cognitive science and educational data mining is the development of accurate models of human learning. On the basic science side, learning and memory are important functions of the human brain that support our ability to flexibly interact with our environment. On the education side, predictive theories of learning may be leveraged by intelligent tutoring systems (ITS) to individually optimize instruction [3, 20].

Perhaps the most influential approach to modeling student learning in the educational data mining literature is "knowledge tracing" [5, 10] whereby the learned mastery of a particular skill or fact is treated as a latent state and the probability that a person's knowledge is in that state is updated in light of observed student behavior. For example, in Bayesian Knowledge Tracing (BKT), each learning unit is assumed to be in one of two discrete states: $\{unknown, known\}$. Each time the student engages in a learning activity, the latent knowledge can transition from the *unknown* to the *known* state with probability l. Performance on a test, quiz, or exercise is conditional on the latent knowledge state, such that being in the *known* state is typically associated with a higher probability of issuing a correct answer than being in the *unknown* state. Using the model, it is possible to infer posterior probabilities of the knowledge state of each learner and skill using Bayes' rule, given the pattern of responses made on various assessments or quizzes. These probabilities are then used to make predictions about learning performance for new students, as well as to design optimized instruction policies.

Research in this area focuses on building more precise models of student learning by, for instance, incorporating factors that reflect individual abilities [41, 21], contextual factors that contribute to errors [6], or models of the exact moment at which a skill is acquired [7]. However, one relatively underexplored question is what types of observable data may be most useful for informing inferences about latent knowledge states during learning. Of particular interest is the idea that many other features besides overt responses might be partially informative. For example, the student's response time to a test question may add additional information about learning alongside correctness [24, 38, 40]. Likewise, patterns of mouse or eye movements during a learning session might help index drifting attention [8, 27].

In this paper we demonstrate that it is possible to integrate indirect neural measurements of brain activity into a cognitive model of learning in a way that 1) can improve prediction of a learner's test performance at a 72 hour delay and 2) allows knowledge tracing without interrupting the learning environment with explicit tests or assessments (which can be distracting or may bias learning).

Although acquiring neural recordings is impractical in most educational settings, the approach of fusing multiple sources of sensor data about individual learners may be a generally useful method for the educational data mining literature. In addition, as we show in our results, such modeling efforts may also feedback to contribute to a better understanding of the neural and cognitive mechanisms that support learning and memory [2, 1, 34, 35]. Finally, as the cost and difficulty of making indirect neural recordings falls (e.g., due to the advent of portable, dry contact electroencephalogram or EEG) the practicality of utilizing such sensors will likely

*D. Halpern and S. Tubridy contributed equally to the project and author order was determined arbitrarily.

increase (c.f., [14]).

We begin by reviewing past work in cognitive neuroscience which has attempted to identify predictive signals of learning and memory processes. Next we describe our approach fusing concepts from knowledge tracing with what is known about the cognitive neuroscience of memory. We then describe a dataset collected from human participants performing a simple second-language learning task while undergoing functional magnetic resonance imaging (fMRI). We compare the predictive power of a variety of models against held-out memory recall data at study-test delays ranging from one day to one week. From the fitted model we then extract the neural signals corresponding to learning in the study period.

1.1 Prior work using cognitive neuroscience methods to predict individual learning

The prediction and optimization of human learning has been a long standing goal of cognitive neuroscience research. On the prediction side, a number of studies have explored the "subsequent memory" paradigm [28, 23, 13, 26]. In these experiments, participants study controlled stimuli such as lists of word pairs while brain signals (such as the blood oxygen-level dependent "BOLD" signal measured via fMRI or event-related potentials, ERPs, assessed with EEG) are recorded. Some time later, participants' memory is tested for the material they saw during study. Accuracy on each memory test item is used to back-sort the neural data recordings into brain patterns associated with successful versus unsuccessful later memory. Regions with a reliable difference in brain activation between these two classes are taken to reflect neural correlates supporting lasting memory formation. Across these studies a coherent set of brain regions have been identified as being involved in human memory formation including the hippocampus and medial temporal lobe, which have long been associated with memory formation on the basis of animal and lesion studies [29, 9].

Building on this work, Fukuda et al. (2015) identified two EEG-based subsequent memory signals and used these to classify study trials in a memory experiment as likely to be remembered (*initially well studied*) or forgotten (*intially poorly studied*). In a subsequent session, participants were allowed to restudy half of the items identified as *initially well studied* and half of the items identified as *initially poorly studied*. A final test then assessed knowledge for all of the items. Of particular interest was the finding that the restudy opportunity most benefitted the *initially poorly studied* items compared to the other items. Importantly, the entire prediction about what was or wasn't well studied was based exclusively on indirect neural recordings for each subject rather than any explicit assessment or test.

The subsequent memory paradigm has been a powerful tool for studying the neural basis of memory. However, the cognitive neuroscience literature does not currently take advantage of the wealth of knowledge about predicting individual learning from the educational data mining and cognitive modeling literatures. For example, classifying brain patterns as forgotten based on a single test fails to account for the possibility of "slippage" (errors in performance of a mastered skill due to chance) which is central to BKT models [10]. Likewise, when an item is not remembered at test it could be for a number of reasons: the item may have been poorly encoded during the study session, or perhaps was well encoded and would have been remembered at an earlier study session but was simply forgotten due to decay or interference. Structured models such as Hidden Markov Models (HMMs) can account for such latent memory dynamics and use them to help improve predictions. The subsequent memory approach is also difficult to apply when learners get repeated study opportunities because of ambiguity about which brain scans should be classified as causally related to the test performance. Finally, the standards for model development within the machine learning and data mining communities is predictive performance on held-out data which is often more difficult than describing statistically reliably patterns within a single data set due to the ability to overfit.

To address these issues, we describe an approach to the simultaneous modeling of behavior and neural recordings in a single knowledge tracing model[1]. Our aim is to demonstrate the value of combining insights from these still somewhat disparate literatures. The approach we take is in some ways similar to work by Anderson and colleagues that has tried to infer from fMRI the mental state of individuals as they engage in complex math problems [2, 1, 4, 42, 33] (see also [34, 35]). While these reports hint at the utility of combining fMRI with probabilistic cognitive models, this prior work does not specifically address the learning and memory issues considered here.

2. THE OMNI DATA SET

The dataset we consider, part of the NSF-funded "Optimizing Memory using Neural Information" (OMNI) project[2], consists of human performance on a cued-recall memory test for a set of Lithuanian-English word translations. The learner's task is to study the word pairs across multiple presentations and then, after a delay, recall the English associate for a presented Lithuanian word.

Starting with a normed set of Lithuanian-English words, we selected 45 translation pairs [19]. During study, participants saw the translation pairs presented one at a time for 4 seconds each with a variable duration inter-trial interval (4s-16s for consistency with event-related MRI timing). Words were presented on a computer screen with the Lithuanian word at the top of the screen and the English translation underneath.

Each word pair was presented five times and no pair was presented for the nth repetition until all words had $n-1$ presentations. Importantly, and in contrast to many psychology studies on the subsequent memory effect, all participants see the same sequence of study items[3]. Immediately

[1] Here we focus on fMRI due to improved spatial resolution, even though other methods (e.g., EEG and skin conductance response), also provide useful signals that correlate with memory performance and could be incorporated into our approach.

[2] http://gureckislab.org/omni

[3] Although the models we apply do not explicitly model inter-item interactions, maintaining a fixed sequence across participants ensures that some of these inter-item effects will be captured in the model parameters we estimate because,

following the study session participants gave judgments of learning (JOLs, [22]): for each pair participants were presented with the Lithuanian and English word and used the computer mouse to indicate on a scale of 0-100 how likely they were to remember the association in one week.

Participants were given either an immediate recall test (0 hours) or returned to the lab approximately 24, 72, or 168 hours after the initial study session (randomly assigned)[4]. During the recall test, participants saw a Lithuanian word presented on the screen and had to type the associated English word. A trial was coded as correct if participants typed the correct English word (allowing for typographic errors) and all other responses were incorrect.

For more efficient estimation of the different model parameters, we conducted a large behavioral experiment outside of the MRI scanner and combined those data with additional observations from participants who performed the same task during MRI scanning (under this view all participants are equally useful but purely behavioral subjects are treated as though their MRI data are "missing" and so estimates of their learning are based on the observed JOLs and recall performance). Each participant (N=189) was tested at one of the four study-test delays. Among the behavioral participants (i.e., no MRI data) the group Ns were 20, 49, 60, and 49 in the 0, 24, 72, and 168 hour study-test delay groups, respectively. All MRI participants (N=21) were tested at the 72 hour delay.

MRI participants underwent an identical study-test procedure as the behavioral participants except they were scanned during the study session. MRI data were collected on a Siemens Prisma 3T at the New York University Center for Brain Imaging. Functional Blood Oxygen-Level Dependent (BOLD) data covering the cortex were acquired at a spatial resolution of 2.5 mm^3 with a 1 second repetition time (TR; the temporal resolution of the fMRI data) and anatomical scans were collected at a spatial resolution of .75 mm^3.

To summarize, the final data set consists of a record for each learner that contains: the pattern of recall attempts for each list item, JOLs collected after the study session for each list item, and, for each MRI participant, the 65x77x73 set of voxel measurements across 2936 time-points describing the BOLD signal recorded with MRI.

Figure 1 shows key features of the behavioral data. Across the four different test delays, memory performance generally drops, likely due to forgetting. Participant performance varied widely from 0 to 100 percent correct. In addition, across participants, average JOLs following study were weakly correlated with performance ($r = [0.43, 0.24, 0.31, 0.55]$ and $p = [0.06, 0.10, 0.004, 3.4e{-}5]$ in the 0h, 24h, 72h, and 168h groups, respectively). Pooling across all participants, the mean JOL correlation with final performance is low but significant, $r = .365, p < 1e{-}7$.

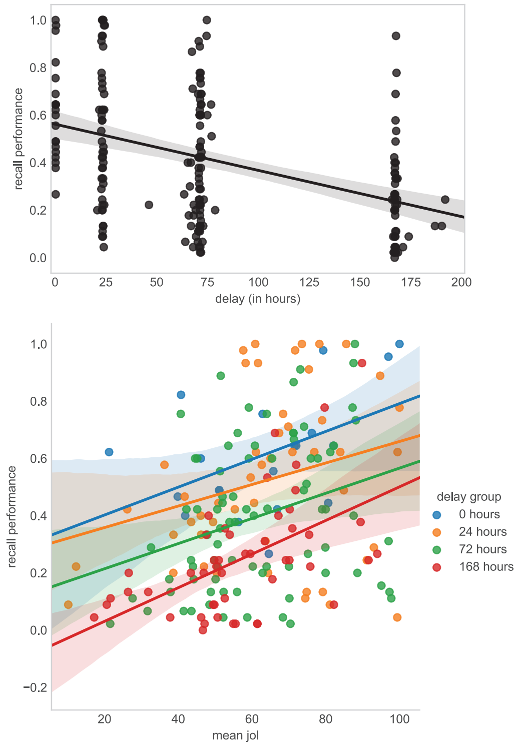


Figure 1: Top: Mean recall performance (% correct) for individuals (dots) at each study-test delay. Bottom: Mean individual participant Judgment of Learning is correlated with individual overall percent recalled within each delay condition.

3. INFERRING KNOWLEDGE STATES FROM BEHAVIORAL AND NEURAL DATA

The following section describes the basic mathematical structure of our models. Similar to BKT, the core of our approach assumes a probabilistic representation of the latent mnemonic status (e.g., *remembered* versus *forgotten*) of each item on the to-be-remembered list and we begin with established two- and three-state models that have shown effectiveness in tracking learning and memory [5, 10]. Where our models differs from past knowledge tracing approaches is that we propose a mapping between these latent mnemonic states and patterns of brain activity that can allow the brain data to inform this inference.

3.1 A Hidden Markov Model of Memory

Like BKT, our approach draws heavily from the structure of HMMs. Each memory trace, i, (i.e., memory for the association between two words) is represented as a non-homogenous, censored Hidden Markov Model with the following properties (notation follows [25]):

3.1.1 States

Each trace can be in one of a number of discrete mnemonic **states**, S. For simplicity we will begin with a two state $S = \{s_U, s_K\}$ model with states corresponding to *unknown* and *known* similar to BKT. However, we also consider a more complex, three-state model first proposed by Atkin-

for instance, the measured difficulty of a word is always assessed with respect to the other list items.

[4] Due to schedule difficulties a one subject returned at 48 hours but we still included their data in the modeling. In addition, 9 of the 72 hour subjects were scanned in a different fMRI scanner but we only include their behavioral data here.

son [5]. The three-state model has states $S = \{s_U, s_K, s_P\}$ corresponding to *unknown*, *known* (with possibility of forgetting), and *permanently known* (see Figure 2). Across both types of models the s_K and s_P states represent memories that have generally higher recall probabilities (e.g., $\Pr[recall = correct|s_P] > 0$), but the s_K state is susceptible to decay between study events while the s_P state is absorbing[5]. The current state of item i at time t will be denoted q_t^i.

3.1.2 Priors

A **prior**, $\pi_{t=0}$, that captures our initial belief of the memory state of all items. The prior for a particular item memory, i, can be written as $\pi_{t=0}^{i,s} = \Pr[q_{t=0}^i = s]$ for $s \in \{s_U, s_K\}$ (two state) or $s \in \{s_U, s_K, s_P\}$ (three state). With unfamiliar learning materials we assume that the initial memory status is heavily biased towards the unknown state (i.e., $\pi_{t=0}^{i,s_U}$ is much higher than for any other state).

3.1.3 Transitions

A set of **transition probabilities**, A, which determine the likelihood that a memory will move between the different states at each time point. In prototypical HMMs the transition probabilities are stationary and the same transitions are applied at each time step. In our model there are different sets of transition probabilities which are applied at a given time step depend on the type of external "event", e_t^i, that occurs (e.g., a study trial versus a time step between trials; Figure 2). For memory trace i the transition probability of moving from state s to s' after an event of type g will be denoted $a_t^{i,g,s\rightarrow s'} = \Pr[q_t^i = s'|e_t^i = g, q_{t-1}^i = s]$ where g indicates the specific event type on trial t.

Event types depend on the particular experiment design but here include "study trial" (*study*), "study with JOL trial" (*study+JOL*), "timestep in which memory decays" (*decay*), and "test trial" (*test*). Generally, during *study* or *study+JOL* events we assume that items tend to transition from a more poorly learned state to a more fully learned state. The probability of transitioning to a new state on a study trial is represented in our three state model by parameters x, y and z and in the two-state model by parameter l (see Figure 2). During *decay*, items in a non-permanent state (s_K) have a probability of transitioning to the *unknown* state with probability f while items in s_P (in the three-state model) remain in the permanently learned state. Decay events are necessary to account for the patterns of forgetting across the study-test delay intervals shown in Figure 1. We assume test trials have no effect on transitions as they appear at the end of the task.

We define an experiment **protocol**, E, as a $\mathcal{N}$ x $\mathcal{T}$ matrix where $\mathcal{N}$ is the number of items being studied and $\mathcal{T}$ is the total number of micro-time steps modeled in the experiment. Each entry of the matrix, e_t^i, codes which of a discrete set of event types occurred on a time step as described above. The protocol captures the dependencies between event sequences that influence different memory traces. For example, if word w is studied on a given trial, then all the other items on the list might undergo a memory decay event during the same time step. This way the protocol enforces the implicit tradeoffs of studying one item over others at a particular point in time.

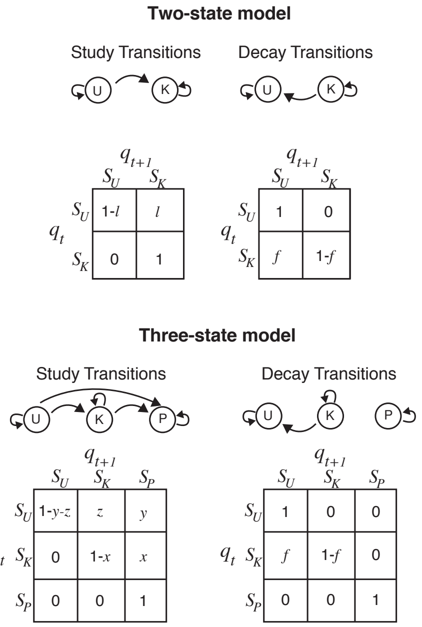


Figure 2: The matrix of transition probabilities for either study or decay events in the two and three state model. The letters within each matrix reflect the transition parameters which are estimated to data. The state labels U are "unknown", K are "known" (with possible forgetting), and P are "permanently known."

3.1.4 Observable signals

The mapping between brain and behavior is made through a set of **observation distributions**, B, which define the probabilities that, on event type g at time t, an observable random variable of data type d, $\mathbf{o}_t^{g,d}$ takes on a value $v_k^{g,d}$ from a (potentially infinite) alphabet $v^{g,d}$. For each memory trace i, we can write the probability of its associated observables as $b_t^{i,g,s,d}(v_k^{g,d}) = \Pr[o^{g,d} = v_k^{g,d}|e_t^i = g, q_t^i = s]$. Observation distributions in effect define the full generative model that links both behavior and neural information to underlying knowledge states. Here we consider three types of observations: behavioral assessments (*recall*), JOLs (*JOL*), and hemodynamic fMRI measurements (*MRI*). However, this approach can easily incorporate many other measures including response time, pupil dilation, EEG measurements, or alternative fMRI signals.

Behavioral Assessments. At certain points during the experiment the protocol might define a memory test event. On these types of trials the subject might be asked to recall a studied item from memory or to recognize it from a list of alternatives. The response given on these trials is treated as an observation associated with this particu-

[5] One way for the model to capture the difference in performance at 24 versus 168 hours is to assume different mixtures of the s_K and s_P states following learning. For example, at 168 hours, traces in s_P state may dominate correct responses.

lar type of event. Specifically, the alphabet is $v^{test,recall} \in \{correct, incorrrect\}$ and $v^{g,recall} \in \emptyset$ for $g \neq test$, reflecting the absence of any recall response on non-test events. The distribution of test question answers about memory trace i from state s at time t, is then $b_t^{i,test,s,recall}(correct) = p_{\text{recall}_s}$ and $b_t^{i,test,s,recall}(incorrect) = 1 - p_{\text{recall}_s}$ where p_{recall_s} is defined (or fitted) for each memory state. For other trial types, i.e. $g \neq test$, $b_t^{i,g,s,recall}(\emptyset) = 1$. So the update to state posterior probabilities on those events is driven by the state transitions. The parameters governing the probability of issuing a correct response conditioned on the latent memory state are equivalent to the "guess" and "slip" parameters in BKT.

Judgments of Learning. JOL responses were only given on the last study trial (a *study+JOL* event). JOL data were included in the model as the raw response/100 to each JOL trial for each person, i.e. $v^{study+JOL,JOL} \in [0,1]$ and null for other trial types. We model the distribution of JOLs as a truncated Gaussian distribution in the range 0 to 1, i.e. $b_t^{i,study+JOL,s,JOL} = TN(\mu_{JOL_s}, \sigma_{JOL_s}, 0, 1)$ with μ_{JOL_s} and σ_{JOL_s} defined independently for each state s.

Hemodynamic fMRI measurement. Functional MRI scans provide time-series data for each of a large set of 3-dimensional voxels tiling the imaged volume (e.g., the brain). In studies measuring fMRI activation levels at specific time-points it is common to estimate the activation level within voxels and then average voxels within spatial clusters, whether spatially contiguous (regions of interest, or ROIs) or sets of spatially disjoint but functionally related voxels showing similar response profiles (e.g., independent components). Due to the central limit theorem we can expect that the mean activation within a set of such voxels will be approximately normal. We also expect, based on prior work, that there will be a mean shift in the fMRI activation levels of various brain regions during study trials that are later remembered compared to those that are later forgotten [13, 26]. We collect fMRI data for each *study* trial. The fMRI observation consists of N_{fMRI} features. Therefore, $v^{study,MRI} \in \mathbb{R}^{N_{\text{fMRI}}}$ and null otherwise. We model the fMRI state observation distributions as independent Gaussians for each feature n_i, i.e. $b_t^{i,study,s,MRI_{n_i}} = N(\mu_{MRI_{n_i}s}, \sigma_{MRI_{n_i}s})$.

3.1.5 Inference

The full model is specified by a protocol, E, a set of priors over the states, $\pi_{t=0}$, a set of transition probabilities, A, and a set of observation distributions associated with each state-event pair, B. Using Bayes' rule, the posterior probability that a memory trace on trial t is in state $s' \in S$ is:

$$\pi_t^{i,s'} = \frac{b_t^{i,g,s',d} a_t^{i,g,s\to s'} \pi_{t-1}^{i,s}}{\sum_j b_t^{i,g,s_j,d} a_t^{i,g,s\to s_j} \pi_{t-1}^{i,s}} \qquad (1)$$

3.1.6 Illustrative calculation

To illustrate the impact of hypothetical fMRI observations, consider Figure 3 which shows the protocol, E, for the timing of study events for two memory traces (Panel A): item 1 (black) and item 2 (white). On time points where item 1 is studied the protocol has a black cell (and similarly for item 2 using white). Panel B shows hypothetical evolutions over time for the two-state posterior probabilities $\{s_U, s_K\}$ for item 1 obtained by applying the study and forgetting transitions as shown in Figure 2 but without other observable information (i.e., a Markov model). In this example we set the l transition parameter applied on study events to 0.4 and the f parameter governing decay to 0.1.

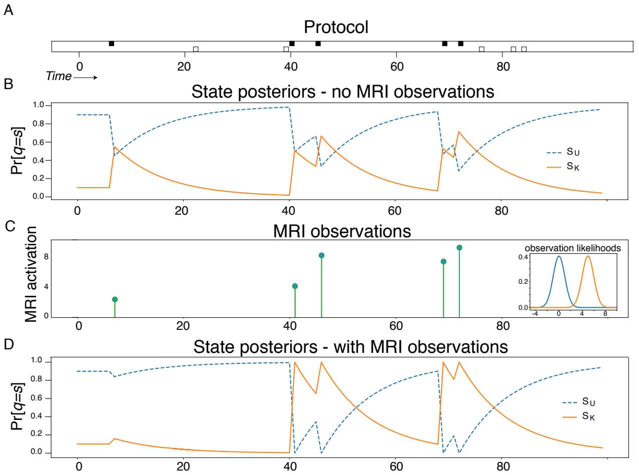


Figure 3: Example illustration of the effect of fMRI observations on inferences about latent knowledge in a two state-model. A) Protocol showing the timing of study events for item 1 (black boxes) and item 2 (white boxes). B) State posterior estimates for item 1 obtained from a hypothetical setting of the two-state model parameters (dashed blue = S_U, solid orange = S_K). C) Hypothetical "observed" fMRI signal on each study trial for item 1 (inset shows the probability density function over MRI observation values conditioned on the state). D) State posteriors for item 1 after incorporating the observation likelihoods from study trials for this item. The inferred state probabilities are dramatically altered by the incorporation of the MRI observation (see text).

At time point 1 the priors reflect the fact that before any study attempts a person is unlikely to know the item (e.g., $\pi_{t=0}^{i=1,s_U} = .9$). At time point 6, item 1 is presented for study for the first time and the posterior probabilities of each state are updated by applying the study transition probabilities to the state posteriors on time $t-1$. Immediately after this study event, Panel B shows that there is now an increased probability of item 1 being in state s_K (solid orange line). However, between time point 6 and 40, item 1 is not presented again and so for each time step between we apply the decay transitions leading to gradual forgetting.

The addition of observable signals that are probabilistically related to latent memory states alters these predictions. The inset figure in Panel C shows how the mean response from a set of voxels in the human brain might result in Gaussian-distributed summed BOLD signals that overlap but differ depending on the state of the memory (e.g., signal being stronger for s_K, orange, than for the s_U, blue, state). Panel C illustrates a hypothetical sequence of fMRI measurements that could be made about item 1 during the study trials (i.e., samples from the Gaussian distributions from the inset plot).

Panel D shows the posterior estimates of item 1's state at each time point obtained through combination of the transition dynamics **and** MRI observations (i.e., using the Hidden Markov Model). As can be seen comparing panel B and D, the addition of observations that are probabilistically associated with latent states can lead to different inferences about the posterior probabilities over those states. Until item 1 is presented at time point 6 the posterior estimates are the same in the Markov and Hidden Markov Models. However, at time point 6 we observe a fMRI signal of a particular magnitude which in turn has a likelihood of originating from each of the two underlying states. If we take into account the observed signal, our estimates of the posterior over states change, since a fairly small signal was observed and the likelihood of such a signal is substantially larger for state s_U than s_K. Consequently, our belief that the item is in state s_K is lower when we include the observation in our estimates than when we simply use the transition probabilities.

Similarly, at time point 40 item 1 is presented for a second study opportunity. Without observations our best estimate of the state probabilities suggests we should be indifferent between s_U or s_K, but the larger MRI observation observed is unlikely to have emerged from the unknown state and so the observation-constrained posterior estimates are weighted much more heavily towards the s_K state. By including the Markov dynamics characterizing the likely temporal evolution of memories, we can adjudicate between otherwise ambiguous neural signals by appropriately dealing with uncertainty in measurement.

3.1.7 Model Evaluation and Fitting Procedure

The following section details the model evaluation, comparison, and feature selection strategies we used.

Model parameterization. Partially due to identifiability concerns [37, 16], some parameters were fixed to semantically coherent values [15], while others were estimated from the data.

For all words we fixed the initial state priors, $\pi_{t=0}$, as $[.99, .01]$ or $[0.99, 0.005, 0.005]$ for s_U, s_K in the two-state model or s_U, s_K, and s_P in the three-state model, respectively. This was motivated by the fact that none of the learners in our dataset had prior experience with Lithuanian. We also fixed the probabilities of giving the correct test response, $\mathbf{p}_{\text{recall}}$ as $[.01, .9]$ and $[.01, .9, .9]$ for latent memory states s_U and s_K (two-state model) or s_U, s_K, and s_P (three state model, see below), respectively. This reflects the assumption that it is very unlikely that one would guess the correct answer in a cued recall test without any memory ($s = s_U$) and that, as in [5], the primary difference between s_K and s_P in the three-state model is the susceptibility to decay over time rather than the availability of a memory to recall (via the influence of the f parameter; see Figure 2).

Fitted parameters include those determining the transition probabilities and observation distributions within each model. Both the two- and three-state models have transition probabilities to fit for each word pair w (summarized in Figure 2). In the two-state model these are the l_w and f_w parameters controlling memory strengthening and decay, respectively. For the three-state models, the x_w, y_w, and z_w values control transitions between states during study opportunities and the f_w parameter determines forgetting rates.

Although the learning trajectories for each word pair were instantiated in separate HMMs, to get better estimates of the parameters we used a hierarchical Bayesian model that used group-level priors over the parameters to regularize the estimates. Each x_w was drawn from a Logit-Normal(x, σ_x) where x itself was drawn from a Normal(0, 6) and σ_x was drawn from a Truncated-Normal(0, 1). The model for the f_w parameters was exactly the same. The simplices zy_w were generated using the following procedure: z and y were drawn from a Normal(0, 6). z_w and y_w were drawn from Normal(z, σ_z) and Normal(y, $sigma_y$) respectively with σ_z and σ_y both drawn from a Truncated-Normal(0, 1). Finally, zy_w was set to $softmax([0, z_w, y_w[)$. This can be thought of as a multivariate generalization of the Logit-Normal with a diagonal covariance matrix.

When fitting models that incorporated JOLs or MRI data we also estimated the means and variance parameters for the Gaussian (truncated for JOLs) observation likelihood from each latent state. For the JOL distributions, each μ_{JOL_s} was drawn from a Normal(.5, .5) and each σ_{JOL_s} was drawn from Inverse-Gamma(1, 2). Similarly, for each fMRI feature n_i (see below) in state s, $\mu_{MRI_{n_i}s}$ was drawn from a Normal(0, 1) and $\sigma_{MRI_{n_i}s}$ was drawn from an Inverse-Gamma(1, 2).

fMRI feature selection. After standard MRI preprocessing [11], we selected data for inclusion in the model. We reduced the dimensionality of the fMRI data using group spatial independent components analysis (ICA) using the ICASSO algorithm as implemented in the GIFT ICA toolbox (http://mialab.mrn.org/software/gift/) [**?**, **?**]. This procedure, which is blind to trial information and memory outcome, resulted in a set of 60 independent components that are characterized by a particular temporal (the timecourse of activation) and spatial (the loading of each component on fMRI voxels) profile for each participant. Components that were unstable across estimations (ICASSO) and components associated with signal from ventricles or motion were discarded leaving 43 independent components for inclusion as model features. Individual trial activations for each identified component were summarized as the mean of timepoints encompassing 4-6 seconds post-stimulus onset (to account for the temporal lag in the BOLD response), resulting in one activation value for each trial in each component for each MRI participant.

Model estimation. We used MCMC sampling via the NUTS algorithm as implemented in Stan [31] to estimate the posterior over the parameters (4 chains of 200 iterations; 100 per chain discarded as burnin; 400 total samples per parameter). To ensure convergence, we checked that estimates of the probability of recall had low $\hat{R}$ values (a measure of whether the sampling chains are converging to similar estimates) [32, **?**].

Model evaluation. In order to compare models, we want to evaluate how well our models will predict new, unseen data. It is generally agreed that the generalization method with the fewest assumptions is leave-one-out cross validation, which is preferred when sufficient data and computational resources are available [39]. To conserve on computational resources, here we use K-fold cross validation, setting K to 10. Because our goal is to assess the utility of incorporating MRI signals into a memory model, the held-out data only included data from the 20 fMRI subjects. We divided up the data from these subjects into ten equally sized folds. We then trained ten versions of each model where the training set consisted of all of the data from behavior-only subjects and nine of the ten folds of the fMRI subjects. On the held-out test set, we used the identity of the words and the trial timings (and JOL or fMRI observations, where appropriate) to generate the posterior probability of recall for each held out word at the time of test.

As we are primarily interested in our ability to classify a new piece of data as successfully recalled or not rather than the log likelihood of the trial under the model, we adopted a cross-validated area under the ROC curve metric (ROC-AUC). The ROC-AUC can be interpreted somewhat like an accuracy measure where 0.5 represents chance prediction and higher values indicate better predictive performance of the model. Using ROC-AUC allows us to compare the held-out predictive performance of models with varying numbers of parameters while providing a metric of model performance that is relatively insensitive to class imbalance and does not prioritize one kind of error over another (e.g., trading off Hits versus Misses). The model ROCs were defined by calculating, in each cross validation fold, the proportion of predicted as remembered trials that were recalled correctly (*Hits*) and the proportion of predicted as remembered trials that were not (*False Alarms*) at each level of posterior recall probability given by the model.

Model Comparison. We fit three variants of each of the two- and three-state models: a *Recall* model fit to trial timing and recall performance (the binary recall success scores for each word); a model fit to trial timing, recall performance, and JOL observations (*Recall+JOL*); and a model fit to trial timing, recall performance, and fMRI observations (*Recall+MRI*). In each case the training data included data from all of the behavioral participants and a subset of the MRI participant data, and models were evaluated on held-out data. The logic of these comparisons is to see if the models incorporating additional observations (*Recall+JOL* and *Recall+MRI*) provide a better basis for prediction than do the purely behavioral models. In addition, we are interested in whether the model incorporating MRI observations is able to outperform the model incorporating JOLs. This would suggest that the brain data contains more information relevant about memory performance than do people's own self-reports about their memory fidelity. While we are ultimately interested in held-out predictive performance, the models do differ in model complexity. In raw numbers, for the two-state models, the *Recall* model had 2 x 45 word parameters and 4 hyperparameters, the $Recall + JOL$ model added 4 parameters, and the $Recall + MRI$ model added $4N_{\text{fMRI}}$ parameters. For the three state models, the *Recall* model had 4 x 45 word parameters and 7 hyperparameters, the $Recall + JOL$ model added 6 parameters, and the $Recall + MRI$ model added $6N_{\text{fMRI}}$ parameters. However, due to the hierarchical nature of these models, the effective number of parameters may have differed depending on the amount of regularization done by the hierarchical prior.

Table 1: Cross validated Area Under the Curve of the Receiver-Operating Characteristic (ROC-AUC) with ± standard error (in parentheses) across folds.

	two-state model	three-state model
Recall	0.64 (.02)	.64 (.02)
Recall+JOL	0.73 (.01)	.73 (.01)
Recall+MRI	0.72 (.02)	.75 (.01)

4. RESULTS

4.1 Two-state model

For each variant of the two-state model (*Recall*, *Recall+JOL*, *Recall+MRI*) we computed the ROC-AUC for predictions of recall accuracy in held-out trials for the MRI participants. The *Recall* model, trained on the timing of study and test trials and recall performance, achieved a mean (across held-out folds) ROC-AUC of 0.64 (±.02), providing an above chance baseline model against which to evaluate the utility of JOL and fMRI observations (Figure 4A).

The *Recall+JOL*, which adds judgments of learning to both the training and evaluation of the *Recall* model, achieved a mean held-out ROC-AUC of .73 (±.01), improving our predictions relative to the *Recall* model. This shows that metacognitive judgments collected from individuals at the end of a learning session can be used to refine predictions about held-out recall performance.

We next assessed whether fMRI signals recorded during study events could be leveraged to make predictions about held-out performance. The *Recall+MRI* model yielded a held-out ROC-AUC of 0.72 (±.02). Although the held-out performance did not surpass the *Recall+JOL* model, this result indicated that there may be information in the MRI measurements that could be used to make predictions about held-out memory recall performance.

4.2 Three-state model

We next considered whether a more elaborated model of memory could leverage more subtle dynamics of the fMRI data.[6] The held out ROC-AUCs for the *Recall* and *Recall+JOL* three-state models did not differ from those observed in the two-state model (Figure 4B). However, the three-state MRI model boosted the held-out AUC to .75 (±.01) which was an improvement compared to the original two-state *Recall+MRI* model. This was also, in terms of held-out predictions, the most successful model we considered in these comparisons (but see Conclusions), building confidence in the utility of incorporating neural signals into knowledge tracing models.

[6] Although our primary interest in this work is evaluating the held-out predictions of our models, we note that complexity of the three-state model means that three-state *Recall* or *Recall+JOL* variants may not be identifiable due to the sparseness of observations (a single recall outcome or the recall outcome and a single JOL) [37, 16] However, for the MRI participants we have data for every trial, enabling estimation of a three-state *Recall+MRI* model.

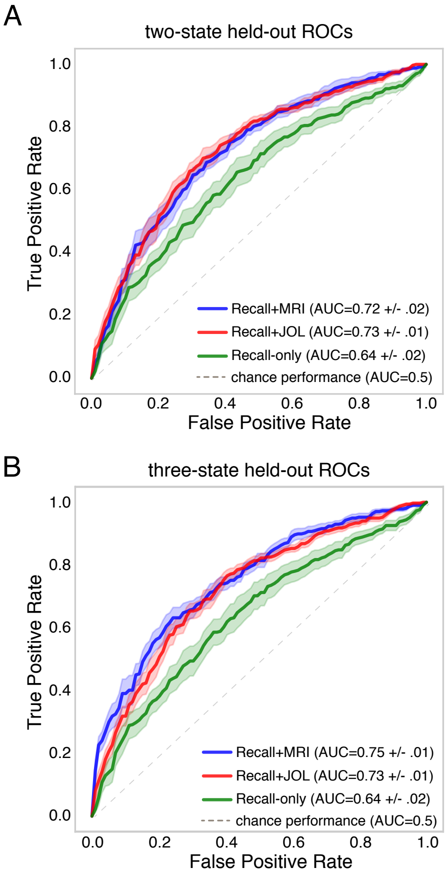


Figure 4: ROC curves for held-out predictions in each of the two-state (panel A) and three-state (panel B) model variants (*Recall*, *Recall+JOL*, *Recall+MRI*). The curves show the mean ± sem across each of the cross validation folds.

In addition, whereas the *Recall* and *Recall+JOL* models did not discriminate between the two- and three-state models, the fMRI data enabled better predictions using the three-state model, highlighting the utility of neuroimaging data in selecting between cognitive models.

4.3 Relating model dynamics to the brain

In addition to the improvements in memory prediction afforded by joint modeling of behavioral and neural data, our approach also allows for examination of fMRI data in light of the estimated models. Figure 5 presents two example analyses in this vein.

Figure 5A shows the contrast map resulting from regressing the change in posterior probability of s_K associated with each study trial (as estimated in the two-state *Recall* model) against the fMRI time-series in each voxel. Using the estimated two-state *Recall* model parameters, we extracted the state posteriors on each study event for the MRI participants based on the sequence and timing of study trials. We then calculated the change in predicted state posterior from just before to just after a study trial and used this change as the predictor for brain activations. This analysis is related to the General Linear Model approach often used in the subsequent memory literature, except that rather than using binary regressors that coded for *remembered* or *forgotten* outcomes as determined by a recall test, we used the estimated continuous state posteriors from the two-state model.

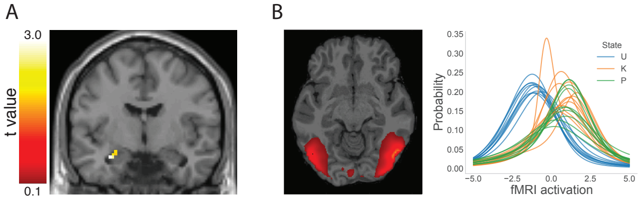


Figure 5: Examples of using estimated model to analyze the brain. A) Coronal slice showing left anterior hippocampal voxels tracking the change in s_K state posterior for each study trial. B) Topography (left; axial slice) and posterior predictive distributions (right) for MRI activations from most informative component in the three-state model. Individual traces show the distributions for each fold of the cross validation

Using a knowledge tracing model in this way to provide estimates of when a particular item is learned during a study sequence with multiple repetitions allows for more sensitive analyses of the brain's relationship to cognitive processes unfolding over extended time. Interestingly, we found that the voxels significantly correlated with the change-in-state-posterior regressor were a cluster in left anterior hippocampus, consistent with the hypothesized role for this region in encoding new information into memory [12].

An alternative way to use the fitted models is to examine the estimated fMRI features' observation likelihoods for each latent knowledge state. The *Recall+MRI* model included activation from a number of independent components as candidate neural features. After estimating the model, the fMRI observation parameters can be used to assess which components provided information about the latent model states. Used in this way, the joint model can be used as a tool for understanding how complex cognitive dynamics, especially those that might not be apparent in a more conventional analysis (e.g., a traditional subsequent memory analysis that only considers activation at the time of study and performance at the time of test), are instantiated in the brain. The most informative component in our model was associated with voxels in lateral occipital and fusiform gyrus regions involved in processing complex visual inputs, as shown in an axial slice through the brain (anterior/posterior of the brain is up/down in the image) in figure 5B. The posterior predictive distributions for component activation conditioned on model state are also shown in figure 5B, and these estimated distributions showed stronger activation for items in the K or P states relative to U.

5. CONCLUSIONS

We evaluate a framework for integrating neuroimaging recordings into a knowledge tracing model. Our approach builds upon recent reports showing robust memory-related signals in the brain. We collected a medium-sized data set of human participants performing a second-language acquisition task both inside and outside a scanner. We then compared a variety of models on their ability to predict held out data for the MRI participants. Our most predictive model was a three-state hidden Markov model that incorporated neural measurements. This is interesting because this model was more predictive than alternative approaches that leveraged participants' self-assessment of their learning (JOLs). One conclusion from this analysis is that there seem to be measurable signals in the brain that index the quality of memory with higher fidelity than people's own introspective access.

We also observed that the use of fMRI measurements enabled discriminating between models that were equivalent when using behavioral data (recall or JOL) alone. Whereas the held-out performance of the two- and three-state models was the same for the *Recall* and *Recall+JOL* model variants, using fMRI data to inform the model estimation revealed an improvement for the three- compared to the two-state model. This result points to the ways in which joint modeling of behavioral and neural data can afford insights into cognitive dynamics that might not be available to researchers focusing on more restricted kinds of data.

Although the results are promising, our assumptions about the fMRI data at this stage are simplistic. For example, our model assumed that the distribution of fMRI signals was stable across time. However, it is well known that fMRI signals often show a pattern of *repetition suppression* [18] where the measured BOLD signal is systematically lower on subsequent presentations of an item. A more sophisticated analysis of the brain may lead to improvements in our models. Another particularly interesting direction is to attempt to model individual learner abilities (c.f., [41, 21]) on the basis of patterns of brain activity given the large variance in overall performance across participants (see Figure 1).

Modifications to the model structure might also improve predictions. As an example, in ongoing work we estimated the three-state *Recall+MRI* model but modeled the fMRI observations as arising from transitions between states rather than from the states themselves (i.e., each fMRI component has a distribution of activations associated with *staying* in a state and another distribution associated with *switching* between states). The three-state version of this *Recall+MRI-Transition* model yielded a held out AUC of 0.77 (±.02), which is our best performing model to date. This shows that there is certainly more signal we can exploit from the data by improving our generative model of the fMRI signal. Attempts to improve the fMRI modeling and explore different model structures are continuing.

We have also illustrated several ways in which this kind of simultaneous modeling approach might feedback to our understanding of the role of the brain in supporting learning and memory. Using a model-based regressor coding for the change in posterior probability of latent knowledge states, we identified a significant effect in a left anterior hippocampus region that is known to be involved in memory formation on the basis of past studies [12]. The similarities between this novel analysis approach and past cognitive neuroscience studies give converging evidence about the hypothesized role of these regions. We also used our estimates of the fMRI observation distributions to examine the relationship between fMRI activation arising from different neural components and the latent knowledge states instantiated in the model(s), which is a novel approach to understanding the way psychological mechanisms or processes may be implemented in the brain.

While we acknowledge the practical limitations of acquiring neuroimaging data in an educational setting – although advances in EEG technology and the established ability to measure subsequent memory signals with EEG may enable such use in restricted settings [17, 14] – overall we believe this work represents an encouraging first step for knowledge tracing approaches that utilize indirect neural information as opposed to explicit tests.

6. ACKNOWLEDGMENTS

This research was supported by NSF grant DRL-1631436 and seed funds from the NYU Dean for Science. We thank Mike Mozer for advice.

7. REFERENCES

[1] J. Anderson. Tracking problem solving by multivariate pattern analysis and Hidden Markov Model algorithms. *Neuropsychologia*, 50(4):487–98, 2012.

[2] J. Anderson, S. Betts, J. Ferris, and J. Fincham. Neural imaging to track mental states while using an intelligent tutoring system. *Proceedings of the National Academy of Sciences of the United States of America*, 107(15):7018–23, 2010.

[3] J. Anderson, A. Corbett, K. Koedinger, and R. Pelletier. Cognitive tutors: Lessons learned. *Journal of the Learning Sciences*, 4(2):167–207, 1995.

[4] J. Anderson, J. Fincham, D. Schneider, and J. Yang. Using brain imaging to track problem solving in a complex state space. *NeuroImage*, 60(1):633–43, 2012.

[5] R. Atkinson. Optimizing the learning of a second-language vocabulary. *Journal of Experimental Psychology*, 96:124–129, 1972.

[6] R. Baker, A. Corbett, and V. Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems*, 2008.

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

[8] D. Bondareva, C. Conati, R. Feyzi-Behnagh, J. Harley, R. Azevedo, and F. Bouchet. 2013. *Artificial Intelligence in Education*, pages 229–238, 2013.

[9] M. W. Brown and J. P. Aggleton. Recognition memory: What are the roles of the perirhinal cortex and hippocampus? *Nature Reviews Neuroscience*, 2(1):51–61, January 2001.

[10] A. Corbett and J. Anderson. Knowledge tracking: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4:253–278, 1995.

[11] J. Danker, A. Tompary, and L. Davachi. Trial-by-trial hippocampal encoding activation predicts the fidelity of cortical reinstatement during subsequent retrieval. *Cerebral Cortex*, 27:3515–3524, 2017.

[12] L. Davachi. Item, context and relational episodic encoding in humans. *Curr Opin Neurobiol*, 16(6):693–700, 2006.

[13] L. Davachi, J. Mitchell, and A. Wagner. Multiple routes to memory: distinct medial temporal lobe processes build item and source memories. *Proc Natl Acad Sci U S A*, 100(4):2157–2162, 2003.

[14] S. Dikker, L. Wan, I. Davidesco, L. Kaggen, M. Oostrik, J. McClintock, J. Rowland, G. Michalareas, J. Van Bavel, M. Ding, and D. Poeppel. Brain-to-brain synchrony tracks real-world dynamic group interactions in the classroom. *Current Biology*, 27(9):1375–1380, 2017.

[15] S. Doroudi and E. Brunskill. The misidentified identifiability problem of bayesian knowledge tracing. In *Proceedings of the 10th International Conference on Educational Data Mining*, 2017.

[16] J. Feng. *Essays on learning through practice*. PhD thesis, The University of Chicago, 2017.

[17] K. Fukuda and G. Woodman. Predicting and Improving Recognition Memory Using Multiple Electrophysiological Signals in Real Time. *Psychological science*, pages 0956797615578122–, 2015.

[18] K. Grill-Spector, R. Henson, and A. Martin. Repetition and the brain: neural models of stimulus-specific effects. *Trends in Cognitive Science*, 10(1):14–23, 2006.

[19] P. Grimaldi, M. Pyc, and K. Rawson. Normative multitrial recall performance, metacognitive judgments, and retrieval latencies for lithuanian-english paired associates. *Behavior Research Methods*, 42:634–642, 2010.

[20] K. Koedinger, E. Brunskill, R. Baker, E. McLaughlin, and J. Stamper. New potentials for data-drive intelligent tutoring system development and optimization. *AI Magazing*, 34(3):27–41, 2013.

[21] R. Liu and K. R. Koedinger. Towards reliable and valid measurement of individualized student parameters. In X. Hu, T. Barnes, A. Hershkovitz, and L. Paquette, editors, *Proceedings of the 10th International Conference on Educational Data Mining*, pages 135–142, 2017.

[22] T. Nelson and J. Dulosky. When people's judgments of learning (jol) are extremely accurate at predicting subsequent recall: The delayed-jol effect. *Psychological Science*, 2:267–270, 1991.

[23] K. Paller, M. Kutas, and A. Mayes. Neural correlates of encoding in an incidental learning paradigm. *Electroencephalography and clinical neurophysiology*, 67(4):360–71, 1987.

[24] P. Pavlik and J. Anderson. Using a model to compute the optimal schedule of practice. *Journal of Experimental Psychology: Applied*, 14(2):101–117, 2008.

[25] L. Rabiner. A tutorial on hidden markov models and selected applications to speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[26] C. Ranganath, M. Johnson, and M. D'Esposito. Prefrontal activity associated with working memory and episodic long-term memory. *Neuropsychologia*, 41(3):378–389, 2003.

[27] M. Rau and Z. Pardos. Add eye-tracking aoi data to models of representation skills does not improve prediction accuracy. In *Proceedings of the 9th International Conference on Educational Data Mining*, pages 622–623, 2016.

[28] T. Sanquist, J. Rohrbaugh, K. Syndulko, and D. Lindsley. Electrocortical Signs of Levels of Processing: Perceptual Analysis and Recognition Memory. *Psychophysiology*, 17(6):568–576, 1980.

[29] W. Scoville and B. Milner. Loss of recent memory after bilateral hippocampal lesions. *J Neurol Neurosurg Psychiatry*, 20(1):11–21, 1957.

[30] X. Shen, F. Tokoglu, X. Papademetris, and R. Constable. Groupwise whole-brain parcellation from resting-state fmri data for network node identification. *Neuroimage*, 15(82):403–415, 2013.

[31] Stan Development Team. PyStan: the python interface to Stan, 2017. Version 2.17.0.0.

[32] Stan Development Team. Stan modeling language users guide and reference manual, 2017. Version 2.17.0.0.

[33] C. Tenison, J. M. Fincham, and J. R. Anderson. Phases of learning: How skill acquisition impacts cognitive processing. *Cognitive Psychology*, 87:1–28, 2016.

[34] B. Turner, B. Forstmann, E. Wagenmakers, S. Brown, P. Sederbefg, and M. Steyvers. A bayesian framework for simultaneously modeling neural and behavioral data. *Neuroimage*, 72:193–206, 2013.

[35] B. M. Turner, C. A. Rodriguez, T. M. Norcia, S. M. McClure, and M. Steyvers. Why more is better: Simultaneous modeling of eeg, fmri, and behavioral data. *NeuroImage*, 128:96 – 115, 2016.

[36] B. O. Turner, J. A. Mumford, R. A. Poldrack, and F. G. Ashby. Spatiotemporal activity estimation for multivoxel pattern analysis with rapid event-related designs. *Neuroimage*, 62(3):1429–1438, 2012.

[37] B. van De Sande. Properties of the bayesian knowledge tracing model. *JEDM-Journal of Educational Data Mining*, pages 1–10, 2013.

[38] W. Van Der Linden. Conceptual issues in response-time modeling. *Journal of Educational Measurement*, 46(3):247–272, 2009.

[39] A. Vehtari, A. Gelman, and J. Gabry. Practical bayesian model evaluation using leave-one-out cross-validation and waic. *Statistics and Computing*, 27(5):1413–1432, Sep 2017.

[40] Y. Wang and N. Heffernan. Leveraging first response time into the knowledge tracing model. *Educational Data Mining*, pages 176–179, 2012.

[41] M. Yudelson, K. Koedinger, and G. Gordon. Individualized bayesian knowledge tracing models. *Artificial Intelligence in Education*, 2013.

[42] Q. Zhang, J. Anderson, and R. Kass. Consistency in brain activation predicts success in transfer. In *Proceedings of the 37th Annual Conference of the Cognitive Science Society*. Cognitive Science Society, Austin, TX, 2016.

Filtered Time Series Analyses of Student Problem-Solving Behaviors in Game-based Learning

Robert Sawyer
Department of Computer Science
North Carolina State University
rssawyer@ncsu.edu

Jonathan Rowe
Department of Computer Science
North Carolina State University
jprowe@ncsu.edu

Roger Azevedo
Department of Psychology
North Carolina State University
razeved@ncsu.edu

James Lester
Department of Computer Science
North Carolina State University
lester@ncsu.edu

ABSTRACT

Student interactions with game-based learning environments produce a wide range of in-game problem-solving sequences. These sequences can be viewed as trajectories through a game's problem-solving space. In this paper, we present a general framework for analyzing students' problem-solving behavior in game-based learning environments by filtering their gameplay action sequences into time series representing trajectories through the game's problem-solving space. This framework was investigated with data from a laboratory study conducted with 68 college students tasked with solving the problem scenario in a game-based learning environment for microbiology education, CRYSTAL ISLAND. Using this representation of student problem solving, we derive the slope of the problem-solving trajectories and lock-step Euclidean distance to an expert problem-solving trajectory. Analyses indicate that the trajectory slope and temporal distance to an expert path are both correlated with students' normalized learning gains, as well as a complementary measure of in-game problem-solving performance. The results suggest that the filtered time series framework for analyzing student problem-solving behavior shows significant promise for assessing the temporal nature of student problem solving during game-based learning.

Keywords

Game-based learning, Problem solving, Time series, Dynamic analysis

1. INTRODUCTION

Game-based learning has shown considerable promise for motivating and engaging students in learning [8]. Game-based learning environments engage students by populating game worlds with believable characters and narrative-driven learning experiences. These environments often feature problem-solving scenarios that give students a high degree of agency and freedom. While engaging for students, this freedom also allows different problem-solving strategies to be pursued to varying degrees of effectiveness. Providing adaptive scaffolding to guide students in following effective problem-solving processes is key to creating effective game-based learning experiences. However, determining how to best scaffold student problem solving in game-based learning environments remains an open research question. Scaffolding effectively requires insight into students' problem-solving processes as well as their individual student characteristics.

In order to devise effective models for adaptive scaffolding in game-based learning environments, it is important to consider how the scaffolds will influence students. The models not only need to account for what support to provide, but also when to provide that support. In other words, the dynamic nature of student problem solving within game-based learning environments should be considered when analyzing the problem-solving behaviors of students. Thus, considering the overall sequence of a student's actions in a game-based learning environment is fundamental to making effective scaffolding decisions, including what a student has done thus far, what their general approach has been, and what cognitive and metacognitive strategies they have been using.

The space of possible problem-solving behaviors within a game-based learning environment can be vast, as students explore, inquire, gather information, and attempt to leverage their knowledge and skills to solve the problem scenario over an extended interaction. In these open environments, providing an exemplar solution path that is known to be effective can serve as a useful reference for students. Domain experts solve complex problems more efficiently than novices [12], and their solutions can serve as valuable points of comparison by students who lack relevant problem-solving expertise. The similarity between an expert solution path and a student solution path can be used to draw inferences regarding the student's trajectory through the open problem-solving space of the game-based learning environment.

In this paper, we present a general framework for analyzing the temporal sequence of student problem-solving behaviors in comparison to expert solution paths in game-based learning environments. The framework consists of filtering student problem-solving actions in a game-based learning environment into a time series representing a student's trajectory through the problem-solving space. We investigate the framework with data collected from student interactions with CRYSTAL ISLAND, a game-based learning environment for microbiology education. To evaluate the framework, we compare several key characteristics of the time series, including a comparison between student trajectories and an expert trajectory, with measures of learning and engagement in game-based learning.

2. RELATED WORK

A growing research base focuses on analyzing problem-solving behaviors of students using summary statistics of student interactions with learning environments. Toth and colleagues clustered summary statistics of students' interactions with a computer-based educational assessment to discriminate between students with different proficiency levels in problem solving [32]. Sawyer et al. used rates of emotions and action units during student interactions with a game-based learning environment to model learning and engagement outcomes [28], while Lalle et al. used eye-gaze measures during student trials with ValueChart, an interactive visualization for preferential choice, to predict student confusion [18]. While successful in using student data to model outcomes important for adaptive learning technologies, these methods did not leverage the sequential structure inherent in student problem solving in advanced learning technologies.

Modelling sequences of student actions has important implications for adaptive learning environments, and it has been approached using both supervised and unsupervised learning methods. Kock et al. modeled sequences of user activities in an e-learning tutor as discrete Markov models, detecting problem-solving styles and learning dimensions about learners by clustering on the trained parameters of the models [17]. They subsequently investigated how these data-driven insights about students can be incorporated into an adaptive learning environment by supporting both individual users and groups of collaborative users. Hidden Markov models (HMMs) have been used widely for sequential student behavior modelling [6, 14]. Beal et al. used HMMs to model the actions of high school students [4]. After fitting HMM parameters for each student, they performed clustering based on the transition matrices of individual students to gain insight into differences in behavior and achievement of the clusters. Hansen and colleagues modeled student session log data by modeling student behaviors as distributions of Markov chains [13]. Bayesian knowledge tracing models use sequences of observations of student performance to create hidden Markov models with binary latent states representing student knowledge [9, 15]. All of this work shares the common approach of modeling student action sequences in terms of probabilistic state transitions. In contrast, our work uses characteristics of student problem-solving sequences encoded as trajectories within the game-based learning environment to predict student learning outcomes measured through pre and post-testing.

Sequence mining techniques have been used to investigate student activity sequences in adaptive learning environments to identify frequent behavior patterns and their evolution over time [16]. Martinez et al. used sequence mining on logs of a collaborative tabletop problem-solving application to examine frequently occurring problem-solving strategies in high and low achieving groups [21]. Perera et al. used trace logs of a collaborative software engineering environment to extract frequent patterns and cluster students using k-means clustering [22]. Another widely used approach is applying pattern mining techniques to logs of user behaviors in web-based learning environments [11, 23]. Our work differs from these approaches by analyzing the paths of student behaviors over full gameplay episodes rather than specific subsequences of behaviors. This approach is taken because a full trajectory and segments of the trajectory provide a comprehensive view of a student's problem-solving process, which is composed of a very long sequence of problem-solving behaviors taken to solve the open-ended game-based learning environment.

Bauer et al. devised solution tree visualizations of user interactions with an open-ended puzzle solving game about protein folding, *Foldit* [3]. They used the visualizations to identify key patterns in problem-solving behavior among high and low performers. Others have used visual data mining on player behavior states, projecting visual representations into a more interpretable visual space [2, 19]. Notably, Liu et al. used state features to collapse complex visualizations and interpret key moments of player behaviors [19]. Our work similarly uses dimensionality reduction to create more interpretable visualizations of player behaviors over time. The primary focus of our work is quantifying the problem-solving trajectories of students in game-based learning environments, and the filtering approach we apply to student action sequences supports creating useful visualizations of the students' solution paths through the problem-solving space. While the calculated slopes and distances are quantities, their geometric interpretation with regard to the problem-solving space are also informative.

Snow et al. used a random walk analysis based on student interactions within a game-based system, iSTART-ME, to plot student trajectories and slopes [30]. They later extended this work through comparisons of student behavior patterns against random walks, revealing that students who behaved in a more deterministic manner exhibited higher quality self-explanations [31]. Our work similarly aims to dynamically analyze student trajectories based on interactions within a game-based learning environment, but it differs in several key aspects. First, our work creates student trajectories of problem-solving behaviors within an open-world game-based learning environment, a more complex space, which requires filtering through dimensionality reduction. Second, our work compares student trajectories to an expert solution path as opposed to a random walk. This comparison is particularly useful for informing the design of adaptive scaffolding functionalities in game-based learning environments. Experts and novices solve problems differently [12, 20], and our work provides an automated framework for characterizing how expert and novice problem-solving paths differ from one another.

3. GAME-BASED LEARNING TESTBED

In this work, CRYSTAL ISLAND, a game-based learning environment for microbiology education, was used as a testbed to explore the problem-solving behavior paths of students and an expert. Students who participated in the study played CRYSTAL ISLAND and completed a pre-test and post-test assessing microbiology content knowledge.

3.1 Crystal Island

CRYSTAL ISLAND integrates science problem solving in a game-based learning environment designed for microbiology education. Students adopt the role of a medical field agent tasked with discovering the source and identity of a mysterious epidemic on a remote island. In order to diagnose the illness, students gather information through conversing with a cast of non-player characters. Reading scientific books, articles, and posters scattered throughout the island provides crucial sources of information about microbiology that students need to diagnose the illness. Students test their hypotheses for the epidemic's source by scanning objects for contamination in the virtual laboratory. Students record findings regarding symptoms and contaminated objects on a diagnosis worksheet. The mystery is solved by submitting a completed diagnosis worksheet with the correct illness, source, and treatment

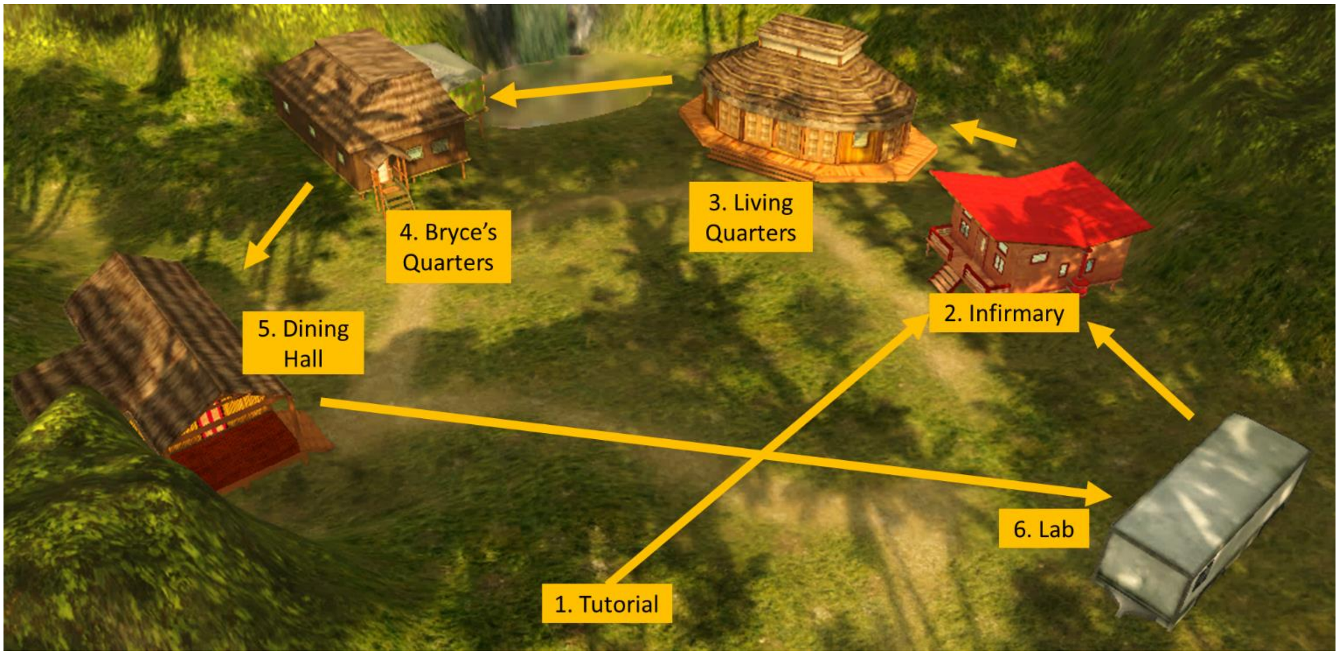


Figure 1. Overview of CRYSTAL ISLAND with expert solution path in gold.

plan to the camp nurse. Throughout solving the mystery, students explore an expansive 3D virtual game environment that includes a beach, infirmary, laboratory, dining hall, and residences.

There are many possible solution paths to solving the mystery successfully. An expert created an expert playthrough for a solution representing a thorough but efficient solution path for the problem-solving scenario. In a related study, a recording of this expert playthrough was used as a *No Agency* condition [7, 29], where students watched the narrated video of the expert solving the CRYSTAL ISLAND problem scenario. The expert visited each building, interacting with each of the virtual characters and reading each of the scientific texts to learn the information needed to solve the mystery (Figure 1). Although it is possible for a student to solve the mystery more quickly by skipping content in the game, the expert playthrough is intended to represent a comprehensive, efficient problem-solving path that any student could implement regardless of prior knowledge. In this work, we analyze students from the *Full Agency* condition of the study, which allowed students to freely explore the game environment after a brief tutorial introducing basic game mechanics. The expert playthrough is used for a comparison of problem-solving behaviors over the course of the gameplay interaction.

The CRYSTAL ISLAND problem scenario consisted of three phases of gameplay: (1) Tutorial, (2) Information Gathering, and (3) Diagnosis. In the Tutorial phase, students learned the basic game controls and mechanics upon arriving on the island's beach. After completing the tutorial, students moved to the main area of the game, beginning the Information Gathering phase. Students gather information through books, posters, and conversing with non-player characters such as the camp nurse, who initiates the game's problem-solving scenario narrative. Students also converse with a range of domain experts and sick patients in the game. Students transition into the Diagnosis phase when they perform their first test with the virtual laboratory scanning equipment. The Diagnosis phase and overall game are solved when students successfully submit their diagnosis worksheet to the camp nurse with the correct illness, contamination source, and treatment plan. Outside of the Tutorial, the phases do not restrict any aspect of a student's experience within the game-based learning environment. The phases are used to segment a student's gameplay for an analysis of problem-solving behavior in different intervals of the scenario.

3.2 Study Participants

The study involved 68 participants from a large mid-Atlantic university who played CRYSTAL ISLAND in a lab setting. After removing students with corrupted data there was a total of 63 students (*M* = 20.1 years old, *SD* = 1.55) of which 42 (66.7%) were female. Prior to interaction with Crystal Island, students completed a 21-question multiple choice pre-test assessing microbiology knowledge (*M* = 11.5 (54.8%), *SD* = 2.7 (13.0%)). Students played for a range of 26.4 to 159.8 minutes (M = 68.0 min, SD = 22.4 min) while the expert playthrough lasted 91 minutes. After completion of the game, students completed the same microbiology assessment as a post-test (*M* = 13.3 (63.5%), *SD* = 2.7 (13.0%)).

3.3 Measures of Learning Performance

A primary goal of CRYSTAL ISLAND is learning relevant microbiology content. We measure student learning in CRYSTAL ISLAND in terms of normalized learning gain, which is the difference between pre and post-test score standardized by the total amount of improvement or decline possible from the pre-test. This calculation uses percentage of questions correct on the pre-test and post-test to calculate learning gain. Students demonstrated positive normalized learning gains with an average normalized learning gain of 0.19 (*SD* = 0.26).

A previously used indicator for in-game student engagement assessing progress and efficiency in the problem-solving scenario is given by *final game score* [25]. This measure was designed to allot points to students for efficient problem-solving behaviors such as talking to key virtual characters and solving the mystery in a short duration while subtracting points for inefficient behaviors such as scanning incorrect items in the virtual laboratory or submitting an incorrect solution. *Final game score* has been shown to be significantly associated with post-test score, independent of pre-test score [25]. Scores varied widely among students with a range of -1543 to 1502 and an average of 673.7 (*SD* = 616). Both learning, as measured by normalized learning gain, and in-game student engagement, as measured by *final game score*, are target learning objectives of game-based learning environments. We therefore investigate how learning and in-game student engagement are related to student problem-solving trajectories in order to evaluate the utility of the filtered time series analysis framework.

4. TIME SERIES ANALYSIS

The similarity of two students over their entire gameplay can be defined as the distance between their trajectories through the game. First, we define student trajectories as filtered cumulative actions over time. Then, we define the temporal distance as the average Euclidean distance between trajectories over each time step, which is known as the *lock-step Euclidean distance* [10]. Distances between students and the expert playthrough are calculated. The slope of the trajectory is calculated as the ordinary least squares regression line through data points of each student's time series, roughly measuring the problem-solving behavior of a student through an adjusted gameplay pace. This distance representing student gameplay similarity to the expert path and regression slope are compared to established measures of learning performance in CRYSTAL ISLAND: normalized learning gain (NLG) and final game score [25].

4.1 Filtering Process

Students perform several different problem-solving behaviors while interacting with CRYSTAL ISLAND. The cumulative counts of student in-game actions are recorded during gameplay, including conversing with virtual characters, reading books and articles, editing the diagnosis worksheet, completing a plot point, submitting a worksheet, and scanning an item in the virtual laboratory. A dimensionality reduction technique to convert the six cumulative counts of actions into a single value describing student progress until a particular moment in time reduces noise in distance measurements by lowering the dimensions used in calculating Euclidean distance. Filtering a multivariate time series to a univariate time series is used in sequential distance methods to reduce the effect of noise on the distance [5].

Due to the correlations between cumulative action counts at specific time intervals, principal component analysis is used for dimensionality reduction [1]. Specifically, the first principal component is used to filter a vector of cumulative action counts at a point in time to a single value (Figure 2). The principal components are calculated on the final action counts of each student (not including the expert counts), and the first principal component (variance explained = 37%) projects the cumulative action vectors onto a single dimension. The first principal component used to filter the cumulative action counts to one dimension is reported in Table 1, along with the means and standard deviations of the final action counts. Table 1 also indicates that the expert solution ("Gold Path") is efficient in terms of the number of in-game actions performed.

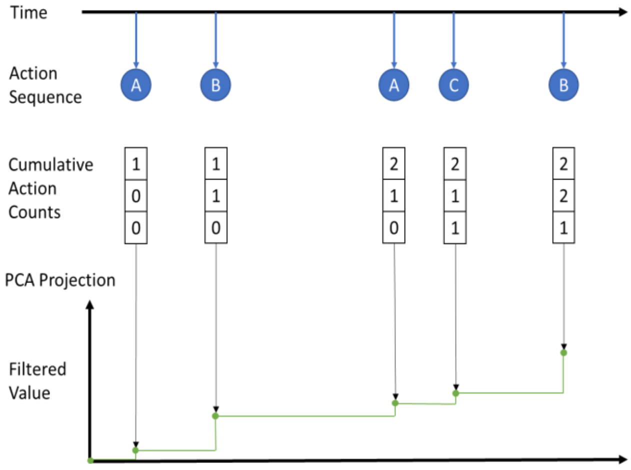


Figure 2. Filtering process from action sequence to time series.

Table 1. Summary statistics of the principal component used for filtering student problem-solving behaviors.

Gameplay Action	First Principal Component	Mean (SD)	Gold Path
Conversation	0.334	18.7 (5.9)	13
Reading	0.554	22.9 (8.0)	21
Worksheet	0.261	24.3 (12.5)	7
Plot Point	0.285	18.7 (1.6)	20
Worksheet Submission	0.444	2.29 (2.6)	1
Scan	0.484	26.0 (16.6)	3

By using this first principal component for filtering, the projection of the cumulative action count vector onto one dimension is guaranteed to be positive and nondecreasing because each element of the principal component is positive, and cumulative action counts are nondecreasing as students play through the game, i.e., as time in game progresses. For example, the transformed gold path final value would be 25.4, and any earlier time has at most the action counts in the final column of Table 1, and would thus have a smaller or equal transformed value. More generally, the filtration can be viewed as a function, *f*, converting the multi-dimensional action vector to a single value, *c*, using the first principal component, **p**. This function is shown in Equation 1 for cumulative action vector **x** of student *i* at time *t*.

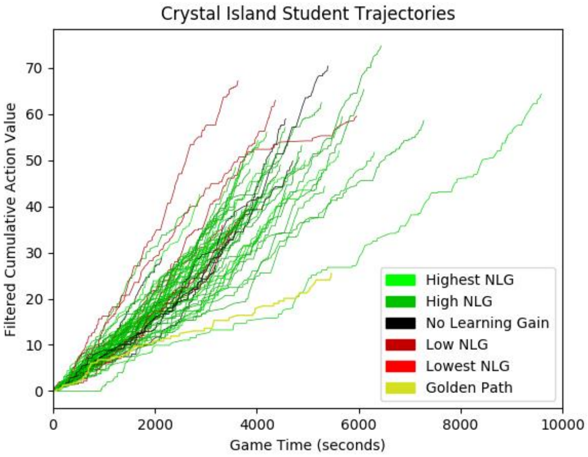


Figure 3. Trajectories of students' interactions in CRYSTAL ISLAND.

$$f(\mathbf{x}_i^t) = (\mathbf{x}_i^t)^{\mathrm{T}}\mathbf{p} = c_i^t \tag{1}$$

A student trajectory is the time series of *c* values, where the time intervals represented by the upper index *t* are flexible. In this work they are calculated for every 10 seconds of gameplay. Figure 3 displays each student trajectory colored by normalized learning gain.

4.2 Trajectory Distance

Once each sequence of cumulative action vectors has been converted to the filtered time series, the lock-step Euclidean distance over the full gameplay session can be calculated. Since students played the game for varying amounts of time, the lengths of each time series may differ. In such cases, when calculating the distance between two series of unequal length, the shorter series is padded to the length of the longer series by repeating the final filtered value. The padding of the shorter sequence prevents violations of the triangle inequality from divergences of two longer sequences with a shorter sequence after the shorter sequence has ended.

The distance between two students is the average Euclidean distance between their filtered time series over all time steps. The average is taken to allow the distances to be compared from different numbers of time intervals. More specifically, the distance, *d*, between students *i* and *j*, can be calculated according to Equation 2, where *n* is the number of time intervals in the longer series. Note that while Minkowski distance of any order would yield equivalent results in this particular case of one dimension, the Euclidean norm is specifically mentioned to generalize to filters with multivariate outputs.

$$d_{ij} = \frac{1}{n}\sum_{t=1}^{n}\left\|c_i^t - c_j^t\right\|_2 \tag{2}$$

The distance between a student's trajectory and the golden path can be calculated by using the golden path as one of the students in Equation 2. The temporal distance calculated by Equation 2 to the golden path for student *i* is denoted g_i. To assess the advantage of taking the trajectory distance, or the average distance over time, a useful comparison is to the final point distance of filtered values, i.e. using only the final time step's filtered value to calculate the distance between students and the golden path. This will allow comparison between similarity measures that take into account the full gameplay over time (Equation 2) and a baseline measure (Equation 3) that does not use the full gameplay session, but instead uses a summary of gameplay. Figure 4 depicts examples of the baseline (a) and temporal distance (b) from one student trajectory to the expert solution path.

$$b_{ij} = \left\|c_i^n - c_j^n\right\|_2 \tag{3}$$

4.2.1 Trajectory Distance per Interval

Since the distance is calculated used a fixed mapping between points in time, the measurement is sensitive to misalignments in time. In other words, *local time shifting*, or similar segments that are out of place, will not be handled by the distance measure [10]. In order to account for similar segments of student trajectories out of place within CRYSTAL ISLAND, the distance over each gameplay phase is calculated. This procedure matches two students' time series from a specific phase to the same start time interval when calculating the distance over that phase, and it uses the same padding procedure described for students with differing phase lengths. Essentially each phase is treated as a "similar segment" and distances are calculated over each phase, matching the start of one student's phase to the start of the other student's similar phase. Figure 4(d) depicts where phases end for two example trajectories, which demonstrate the start points that are matched to calculate phase-based measures.

4.3 Slope of Trajectory

The slope of a trajectory gives important insights regarding the style of problem-solving behavior of students over the course of their interaction with the game-based learning environment. Since the x-axis in this case is time, and the y-axis a filtered measure of cumulative actions, the slope represents the change of the filtered measure of cumulative actions over time. The student's slope can be viewed as a "pace of problem-solving actions," where each problem-solving action's contribution to the pace is weighted by the principal component used to project the cumulative action vector to a single dimension. For example, a student who scans many objects over a specific time span will have a steeper slope in their trajectory than a student who opens their worksheet the same amount of times over that same time interval because scans contribute more to the filtered value than worksheet opens.

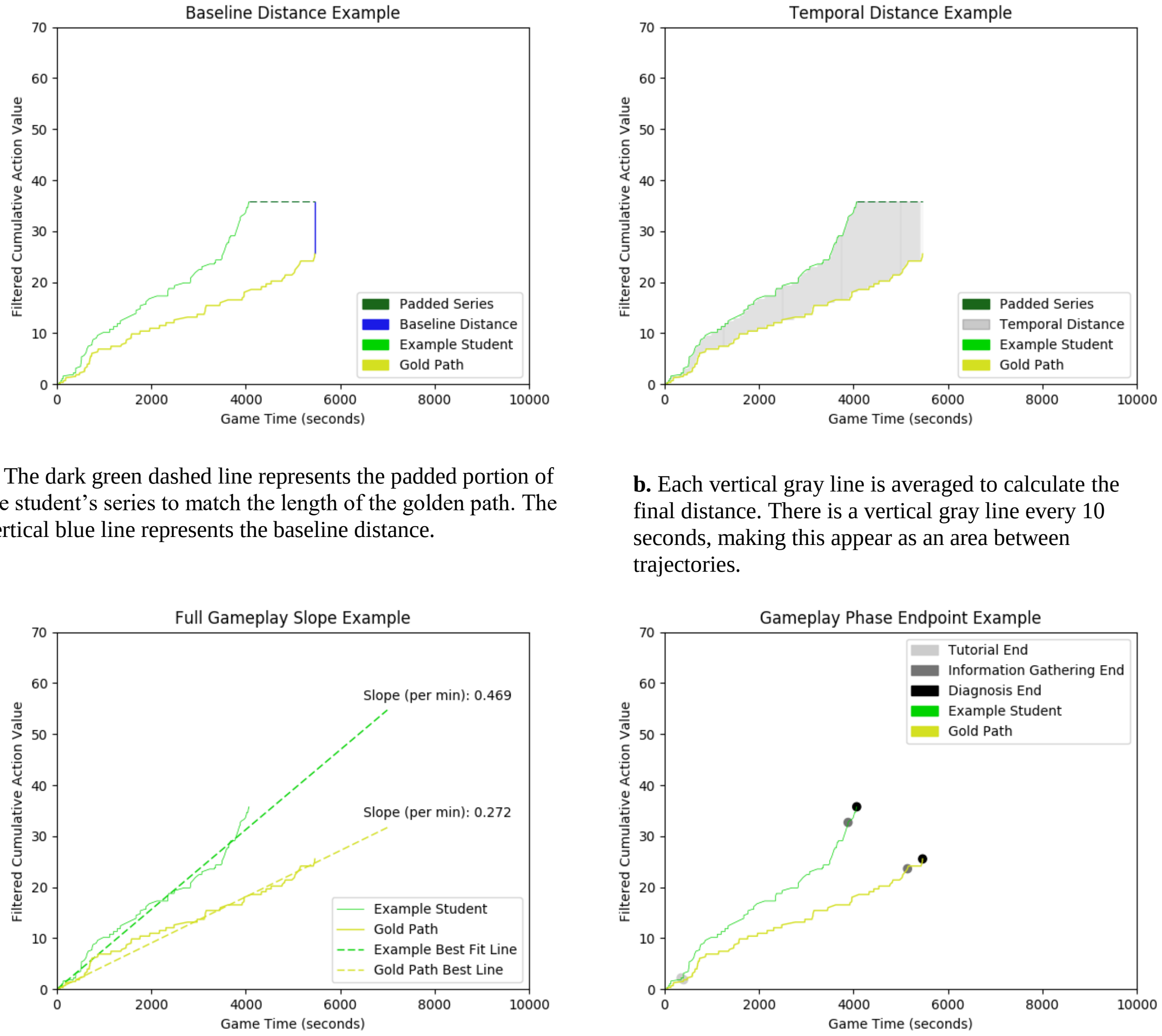
A student trajectory's slope is estimated by fitting a simple linear regression with time (in minutes) as the single predictor of filtered cumulative action value. This is done by using the pairs of points (t, c^t) that create each trajectory of Figure 3 to estimate a line of best fit per student. When fitting the line of best fit over the entire gameplay or Tutorial phase, the intercept is set to 0, since students enter the game with no actions taken. In these cases, the line of best fit is given by $c = \beta\, t$ where *c* is the filtered cumulative action value, *t* is time in minutes, and β is the slope of the student's trajectory. In the Information Gathering and Diagnosis phase, in which a student enters with actions previously taken, the regression line includes an intercept term, $c = \beta\, t + b$, but the slope is the quantity of interest, which has a semantic interpretation as the pace of problem-solving behavior over that time interval.

5. RESULTS

This section analyzes key relationships between students' time series and measures from CRYSTAL ISLAND. First, the relationship between the slope of a trajectory and learning is demonstrated at both a full gameplay level and gameplay phase level. Second, the distance between the gold path and students is analyzed and compared to learning performance in CRYSTAL ISLAND. Third, an analysis of the measures against duration of gameplay is performed to evaluate the independence of the time series analysis against the length of the series. All reported correlations are Pearson product-moment correlations.

5.1 Trajectory Slope Relationship with Learning

A line of best fit through the pairs of time and filtered values were fit to each trajectory as described in Section 4.3. In addition to the line of best fit over the full trajectory (All), lines of best fit were calculated for each gameplay phase (Tutorial, Information Gathering, and Diagnosis). Since the filtered action value is calculated as a weighted sum of cumulative actions, the slope of the line of best fit can be viewed as an estimate of the pace of play of a student within the game-based learning environment with certain actions counting towards the pace more than others. It is also important to note that these slopes are independent of the golden path, but could be compared with cosine similarity as a measure independent of the duration of play. The slopes are found to be marginally significantly correlated with normalized learning gain



a. The dark green dashed line represents the padded portion of the student's series to match the length of the golden path. The vertical blue line represents the baseline distance.

b. Each vertical gray line is averaged to calculate the final distance. There is a vertical gray line every 10 seconds, making this appear as an area between trajectories.

c. The slopes over the full gameplay episode for a student (green) and expert solution (gold).

d. Gameplay phase endpoints plotted in grayscale along a student's trajectory (green) and expert trajectory (gold), illustrating the potential for local time shift issues in calculating distance.

Figure 4. Visual summaries of each time series characteristic calculated for comparison with measures of learning and engagement.

and have a positive cross validation R^2 indicating the generalizability of the results. The results by gameplay phase are reported in Table 2.

When analyzing the simple linear regression leave one out cross-validation R^2 measures, it is important to consider the difficulty of predicting normalized learning gain from in-game actions. More concretely, a baseline using a multiple linear regression using each cumulative action count with game duration (the features used in extracting the trajectory and slope) gives a leave-one-out cross validation R^2 of -0.089. Note that a negative cross-validation R^2 indicates the model predictions on the held-out points have a higher mean squared error than using the variance of the data and are an indicator of poor fit.

Table 2 indicates a relationship between the slope of a trajectory and normalized learning gain. The Tutorial phase is a notable exception here, which indicates that the pace of actions during the Tutorial is not predictive of normalized learning gain. A marginally significant negative correlation between Information Gathering, Diagnosis and slope over the full gameplay session (All) with normalized learning gain demonstrates that as a trajectory slope becomes steeper, the normalized learning gain decreases. This relationship is further exemplified by the positive cross-validation R^2 results, especially relative to the baseline using the cumulative game actions and duration. Thus, a slower pace (lower slope) of students' problem-solving behaviors measured by the filtered cumulative actions in phases beyond the Tutorial are indicative of positive learning outcomes in CRYSTAL ISLAND.

The slope of the expert solution path is the lowest observed slope of any trajectory in the dataset (0.27, next lowest = 0.31). The low slope indicates a relatively slow pace of play in terms of the number of actions taken within the game, which reflects the expert's deliberate and efficient on-task problem-solving path. The deliberate play demonstrates positive problem-solving strategies, such as reading texts thoroughly and planning the next action.

Table 2. Summary of the relationship between trajectory slopes and normalized learning gain.

Gameplay Phase	Average Slope (SD)	Correlation with NLG (p-value)	Simple Linear Regression CV R^2
All	0.51 (0.11)	-0.22 (0.09)	0.0172
Tutorial	0.12 (0.08)	-0.063 (0.62)	-0.0362
Information Gathering	0.49 (0.11)	-0.22 (0.08)	0.0165
Diagnosis	0.58 (0.11)	-0.24 (0.05)	0.0275

5.2 Golden Path Distance Relationship with Learning

The temporal distance between the expert solution path and student trajectories was calculated as in Equation 2. There appears to be a relationship between learning, as measured by normalized learning gain, and similarity of a student trajectory with the golden path. The correlations by gameplay phase between normalized learning gain and gold path distance are given in Table 3. The leave-one-out cross-validation R^2 from a simple linear model using the distance as the lone predictor of normalized learning gain is also given for a measure of generalization of the correlational relationship.

Table 3. Summary of temporal distance between students and expert with normalized learning gain.

Gameplay Phase	Average Distance (SD)	Correlation with NLG (p-value)	Simple Linear Regression CV R^2
All	9.98 (4.0)	-0.23 (0.07)	0.0202
Tutorial	0.76 (0.22)	0.0061 (0.96)	-0.0781
Information Gathering	10.5 (4.8)	-0.20 (0.11)	0.0021
Diagnosis	17.3 (12.0)	-0.13 (0.42)	-0.0206

As seen from Table 3, the negative correlation between distance and normalized learning gain indicates that as student trajectories become farther from the golden path (the distance over time increases), their normalized learning gains decreases. The difference between phases is interesting to note, as the Tutorial phase and Diagnosis phase are not significantly correlated with normalized learning gain, while the Information Gathering phase demonstrates a correlation approaching significance and positive cross-validation R^2 superior to the baseline. The superiority of using the full gameplay for the distance calculation in Table 3 indicates that the time warping problem common among time series analysis may not be an issue in game-based learning. This is likely due to the freedom that game-based learning environments provide students, making recalibration of time intervals difficult to compare amongst students' actions.

5.2.1 Comparison with Baseline Distance

While the relationship between the distance measure incorporating the full gameplay from the gold path and normalized learning gain is encouraging, the necessity of using temporal distance can be assessed by comparing the gold path baseline distance from Equation 3 with normalized learning gain. No significant correlation is observed between the baseline distance from the gold path with normalized learning gain ($r(61) = -0.153$, $p = 0.23$). A baseline comparison using each student's final filtered cumulative action value as a single predictor in an ordinary least squares regression evaluated using leave-one-out cross-validation gives an R^2 of -0.0075. The lack of relationships demonstrated with the baseline distance compared to the correlation of the temporal distance indicates that using the distance from the expert solution over the full gameplay session provides valuable information for predicting normalized learning gain.

5.3 Comparison with Final Game Score

The *final game score* is an in-game measure designed by domain experts specifically for the CRYSTAL ISLAND game-based learning environment to assess student engagement [25]. Thus, comparisons with the *final game score* provide a complementary comparison to normalized learning gain from the actions in CRYSTAL ISLAND to gauge a student's experience. First, it should be noted that a marginally significant positive correlation was observed between normalized learning gain and *final game score* ($r(61) = 0.25$, $p = 0.05$), indicating that students with a high *final game score* have higher normalized learning gains. The magnitudes of the correlations observed with the slope and expert solution distance are similar to the correlation observed between *final game score* and normalized learning gain, despite *final game score* being a hand-crafted measure of performance in CRYSTAL ISLAND while the trajectories are automatically created from student data. This is also seen when comparing the leave-one-out cross-validation R^2 of using *final game score* as the sole predictor in a simple linear regression model, which yields a 0.0265 value when predicting normalized learning gain.

Table 4. Summary of time series characteristics with final game score.

Condition	Slope-based Linear Regression CV R^2	Distance-based Linear Regression CV R^2
All	0.0091	0.28
Tutorial	0.030	0.028
Information Gathering	0.021	0.28
Diagnosis	0.064	0.51

The golden path reflects a trajectory with desirable problem-solving behaviors according to the *final game score* as the expert takes an efficient solution path. For example, the expert uses far less scans of irrelevant virtual objects and incorrect worksheet submissions than the average student, both of which are penalized by the *final game score* for being indicative of guess-and-check behavior. This can be observed by the strong predictive power of the temporal distance to the expert solution path over the *final game score* given in Table 4. These results are notably strong when compared with the slope of the trajectories, which has weaker predictive power over student in-game engagement as measured by *final game score*. The relationship between distance to the expert solution and *final game score* increases as students progress through the phases of CRYSTAL ISLAND. This is likely because students perform actions that more directly impact the *final game score* (scans and worksheet submissions) during the final Diagnosis phase, which is captured by taking the distance over this interval.

6. DISCUSSION

In this work, students' problem-solving behaviors in Crystal Island were transformed into time series representing their trajectories through the problem-solving space. This section provides explanations, considerations, and implications of the results from comparing characteristics of these trajectories with learning outcomes.

6.1 Trajectory Slope

The results suggest that the slope of a student's problem-solving trajectory contains valuable information about their approach to problem solving in the game-based learning environment. The slope of a student's problem-solving trajectory was found to be marginally predictive of normalized learning gain using the full gameplay, Information Gathering phase, and Diagnosis phase. Negative slopes were found to be predictive of higher learning gains, indicating that students who performed more problem-solving actions (weighted through the principal component) per minute had worse learning outcomes.

While the slopes were calculated independently of the expert solution, it is interesting to note that the expert solution had the most gradual slope of any problem-solving trajectory. Therefore, the cosine similarity of best fit lines through trajectories would yield similar results to the current analysis of the slopes, which is independent of the expert solution because steeper slopes would be more dissimilar. Thus, the cosine similarity in this particular context would be analogous to subtracting a constant from each slope, which would not affect the measures used for the analysis in this work. Since these slopes are based on univariate time series, there is no additional information that an analysis of the cosine similarity would provide over an analysis of the slopes themselves. However, the current expert path is only one possible problem-solving solution through this space, and in future work it would be informative to conduct an analysis using solution paths that vary by problem-solving strategy, including negative solution paths, such as a guess-and-check methodology.

The slope during the Information Gathering phase was negatively correlated with learning outcomes. This is an interesting observation given the nature of the Information Gathering phase, where students do not perform any scans in the virtual laboratory. (If they had performed scans, they would be considered to be in the Diagnosis phase). While the steeper slopes indicate a problem-solving strategy more in line with a guess-and-check method, this phase by definition does not include guesses through the scanner. This indicates that the slope of the trajectory includes additional information over identifying potential guess-and-check strategies. A more gradual slope in the Information Gathering phase could be caused by students who are more deliberate in fully reading and comprehending their conversations and reading materials, which would contribute to the negative relationship between trajectory slope and learning outcomes in this phase. This observation is in line with previous research on CRYSTAL ISLAND, which found that information gathering prior to hypothesis generation was correlated with improved problem-solving efficiency [26].

The weak relation between slope trajectory and *final game score* is surprising given the way *final game score* and the filtered cumulative action counts are calculated. *Final game score* penalizes incorrect scans in the virtual laboratory and incorrect worksheet submissions, which are both actions weighted heavily in the filtered cumulative action count. Thus, one would expect a steeper slope to indicate a lower *final game score* since the steep slope indicates problem-solving behaviors likely to have a negative impact on *final game score* being performed at a quicker rate than other students. However, this may be offset by the *final game score* rewarding problem-solving efficiency, which would be indicated by a steeper slope.

6.2 Distance from Expert Solution

The results have important implications regarding the temporal distance of a student's problem-solving trajectory and the expert solution problem-solving trajectory. Since this distance represents the dissimilarity of the student's problem-solving path over time relative to an expert's, the negative correlations between dissimilarity and learning outcomes are as one would expect: as a student's problem-solving path becomes more similar to the expert solution, the student's learning outcomes are expected to be higher. Thus, the results suggest that analyzing a student's problem-solving path in game-based learning with respect to an expert's problem-solving path can yield insight into student learning outcomes, which are measured outside of the game-based learning environment. Interaction with CRYSTAL ISLAND centers on solving a complex problem with multiple solution paths, and the expert solution represents one of many possible paths. Further work should be done in evaluating student solution paths in the context of multiple expert solution paths.

The differences between the temporal distance measure and baseline measure indicate that the temporal distance incorporates additional information regarding the problem-solving behavior path. The baseline distance does not capture information regarding intermediate steps of the problem-solving path, which are critical to learning. This is analogous to only checking if a student obtained the correct answer to a problem without considering the steps the student took to solving the problem. In the context of an ill-structured problem, the temporal distance supports a comparison between the steps students took over the course of gameplay with an expert solution rather than merely considering the final summary statistics of a student.

6.3 Heteroskedasticity of Trajectories

The current filtered cumulative action count provides several benefits such as its interpretability as a nondecreasing measure of weighted problem-solving behaviors performed. However, the trajectories become more dispersed as students follow different problem-solving paths through the game. The wide dispersion is a consequence of the open-ended nature of CRYSTAL ISLAND, which has many valid solution paths defined by trajectories. While this dispersion of trajectories is important for revealing the divergence of problem-solving paths among different students, the dispersion as time increases indicates heteroskedasticity in the filtered values, or an increase in variance among the filtered cumulative action values per time step.

This can be observed in Table 3, where the standard deviation of the distance from the expert solution increases per gameplay phase.

For example, in the Information Gathering phase, the standard deviation of the 63 student trajectory distances from the expert solution is 4.8, and this more than doubles to 12.0 in the Diagnosis phase. Future work should address whether this heteroskedasticity is desired in calculating similarities from distances or whether a variance-adjusted distance would be more appropriate to account for how the population of trajectories become more dispersed as time progresses. For example, the increased variance of distance in later phases may be the cause of the expert distance during Information Gathering being significantly predictive of normalized learning gain while the Diagnosis phase has no predictive ability over normalized learning gain. On the other hand, the distance between students and the expert path in the Diagnosis phase explains more the variance of the *final game score* than the Information Gathering phase, indicating that the wide dispersion of filtered values does not have a negative impact on the relationship between expert distance and *final game score*.

6.4 Implications of Time Series Analysis

The primary result of this work is that the trajectory of a student through the problem-solving space of a game-based learning environment has a relationship with the measured learning outcomes of normalized learning gain and significant relationship with *final game score*. The framework for creating these trajectories is generalizable to game-based learning environments tracking cumulative game actions of students as well as a broad range of advanced learning technologies that support multiple problem-solving paths. Importantly, this includes transforming an expert problem-solving solution path into the same problem-solving space as student paths, and quantifying the similarity of a student solution path relative to the expert solution. While this one expert path represents only one possible solution path through the problem-solving space, this similarity predicts normalized learning gain, indicating the potential for evaluating a student's entire problem-solving path in an open-ended game-based learning environment. The measures used here were shown to be predictive of learning outcomes, but further analysis should be done to determine qualitative characteristics related to learning and self-regulatory processes.

These observations have important design implications for adaptive learning environments. For example, the results suggest that one approach to improving student learning would involve an adaptive learning environment scaffolding a student's problem solving to increase the probability that the student follows a trajectory more closely related to an expert problem-solving path. In the context of a reinforcement learning-based tutorial planner [24, 27], characteristics of the trajectory defined by the filtered cumulative action value could be used as continuous state variables. This work has shown the problem-solving trajectory slope and distance to an expert solution are related to learning and in-game student engagement, suggesting that problem-solving trajectory slope and distance to an expert solution are useful variables to include in a state representation for a tutorial planner. The impact of decisions made by the tutorial planner on the student's trajectory in terms of its slope and distance from an expert solution could thereby be used as estimates for the transitions of a decision in a model-based reinforcement learning framework.

These results also have another key implication for the design of adaptive learning environments. In a recent study with the CRYSTAL ISLAND game-based learning environment, students who followed a predetermined path achieved significantly higher normalized learning gains than students who had freedom of control [29]. These results suggest a possible explanation for the higher observed learning gains: students on the predetermined path followed a problem-solving trajectory more similar to the expert solution path than students who were given freedom to explore. Therefore, the effectiveness of an expert solution path could be measured using this framework for time series analysis of problem-solving behaviors, and the solution path could be considered for a limited agency design of a game-based learning environment.

7. CONCLUSION

Open-ended game-based learning environments allow a wide range of problem-solving behaviors. Analyzing student actions within a game-based learning environment can thus provide insight into students' learning processes. Incorporating the sequential nature of student actions within the game-based learning environment is important because of the complexities of the problem-solving process. This work addresses these issues by examining the dynamics of problem-solving behavior of students within a game-based learning environment through a filtered time series analysis.

A general framework for filtering problem-solving behaviors into a gameplay trajectory was presented using a dimensionality reduction filter. The slope of this trajectory, representing the pace of problem-solving behaviors, was shown to be negatively correlated with learning, indicating that students who were more deliberate in the rate of problem-solving behaviors achieved higher learning gains. The similarity of student problem-solving trajectories with an expert solution was shown to be correlated with learning, indicating students who took a similar solution path to the expert demonstrated higher learning gains. A comparison of temporal distance, using the sequential nature of the problem-solving process, and a baseline distance, using a final summary of student problem-solving process, demonstrated the utility of incorporating the temporal nature of interactions within a game-based learning environment. The results demonstrate the value of analyzing the characteristics of a student's path through the problem-solving space in the context of an expert path. In future work, it will be important to investigate how the results of time series analyses can most effectively inform runtime learning environment adaptations.

8. ACKNOWLEDGMENTS

We would like to thank our collaborators in the Center for Educational Informatics and the SMART Lab at North Carolina State University. This study was supported by funding from the Social Sciences and Humanities Research Council of Canada. Any conclusions in this material do not necessarily reflect the views of the SSHRC.

9. REFERENCES

[1] Abdi, H. and Williams, L. 2010. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*. 2, 4, 433–470.

[2] Andersen, E., Liu, Y., Apter, E., Boucher-Genesse, F. and Popović, Z. 2010. Gameplay analysis through state projection. *Proceedings of the 5th Int. Conference on the Foundations of Digital Games*, 1–8.

[3] Bauer, A., Flatten, J. and Popovic, Z. 2017. Analysis of problem-solving behavior in open-ended scientific-discovery game challenges. *Proceedings of the 10th Int. Conference on Educational Data Mining*, 32–39.

[4] Beal, C., Mitra, S. and Cohen, P. 2007. Modeling learning patterns of students with a tutoring system using hidden

Markov models. *Proceedings of the 13th Int. Conference on Artificial Intelligence in Education*, 238–245.

[5] Box, G., Jenkins, G. and Reinsel, G. 1994. Time Series Analysis - Forecasting and Control. *Prentice Hall New Jersey 1994*. SFB 373, Chapter 5, 837–900.

[6] Boyer, K., Phillips, R., Ingram, A., Ha, E., Wallis, M., Vouk, M. and Lester, J. 2011. Investigating the relationship between dialogue structure and tutoring effectiveness: A hidden markov modeling approach. *International Journal of Artificial Intelligence in Education*. 21, 1–2, 65–81.

[7] Bradbury, A., Taub, M. and Azevedo, R. 2017. The effects of autonomy on emotions and learning in game-based learning environments. *39th Annual Meeting of the Cognitive Science Society*, 1666–1671.

[8] Clark, D., Tanner-Smith, E. and Killingsworth, S. 2016. Digital games, design, and learning: A systematic review and Meta-Analysis. *Review of Educational Research*. 86, 1, 79–122.

[9] Corbett, A. and Anderson, J. 1995. Knowledge-tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User Adopted Interaction*. 4, 253–278.

[10] Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E. 2008. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*. 1, 2, 1542–1552.

[11] Ha, S., Bae, S. and Park, S. 2000. Web mining for distance education. *Proceedings of the IEEE Int. Conference on Management of Innovation and Technology*, 715–719.

[12] Hambrick, D., Burgoyne, A., Macnamara, B. and Ullén, F. 2018. Toward a multifactorial model of expertise: beyond born versus made. *Annals of the New York Academy of Sciences*, 1–12.

[13] Hansen, C., Hansen, C., Hjuler, N., Alstrup, S. and Lioma, C. 2017. Sequence modelling for analysing student interaction with educational systems. *Proceedings of the 10th Int. Conference on Educational Data Mining*, 232–237.

[14] Jeong, H. and Biswas, G. 2008. Mining Student Behavior Models in Learning by Teaching Environments. *Proceedings of the 1st Int. Conference on Educational Data Mining*, 127–136.

[15] Khajah, M., Lindsey, R., and Mozer, M. 2016. How deep is knowledge tracing? *Proceedings of the 9th Int. Conference on Educational Data Mining*, 94–101.

[16] Kinnebrew, J., Segedy, J. and Biswas, G. 2014. Analyzing the temporal evolution of students' behaviors in open-ended learning environments. *Metacognition and Learning*. 9, 2, 187–215.

[17] Köck, M. and Paramythis, A. 2011. Activity sequence modelling and dynamic clustering for personalized e-learning. *User Modeling and User-Adapted Interaction*. 21, 1–2, 51–97.

[18] Lallé, S., Conati, C. and Carenini, G. 2016. Predicting confusion in information visualization from eye tracking and interaction data. *Proceedings of the 25th Int. Joint Conference on Artificial Intelligence*, 2529–2535.

[19] Liu, Y., Andersen, E. and Snider, R. 2011. Feature-based projections for effective playtrace analysis. *Int. Conference on Foundations of Digital Games*, 69–76.

[20] Malkiewich, L., Baker, R., Shute, V., Kai, S. and Paquette, L. 2016. Classifying behavior to elucidate elegant problem solving in an educational game. *Proceedings of the 9th Int. Conference on Educational Data Mining*, 448–453.

[21] Martinez, R., Yacef, K. and Kay, J. 2011. Analysing frequent sequential patterns of collaborative learning activity around an interactive tabletop. *Proceedings of the 4th Int. Educational Data Mining*, 111–120.

[22] Perera, D., Kay, J., Koprinska, I., Yacef, K. and 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.

[23] Romero, C. and Ventura, S. 2007. Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*. 33, 1, 135–146.

[24] Rowe, J. and Lester, J. 2015. Improving student problem solving in narrative-centered learning environments : A modular reinforcement learning framework. *Proceedings of the 17th Int. Conference on Artificial Intelligence in Education*, 419–428.

[25] Rowe, J., Shores, L., Mott, B. and Lester, J. 2011. Integrating learning, problem solving, and engagement in narrative-centered learning environments. *Int. Journal of Artificial Intelligence in Education*. 21, 1–2, 115–133.

[26] Sabourin, J., Rowe, J., Mott, B., Lester, J., Carolina, N. and Carolina, N. 2012. Exploring inquiry-based problem-solving strategies in game-based learning environments. *Proceedings of the 11th Int. Conference on Intelligent Tutoring Systems*, 470–475.

[27] Sawyer, R., Rowe, J. and Lester, J. 2017. Balancing learning and engagement in game-based learning environments with multi-objective reinforcement learning. *Proceedings of the 18th Int. Conference on Artificial Intelligence in Education*, 323–334.

[28] Sawyer, R., Smith, A., Rowe, J., Azevedo, R. and Lester, J. 2017. Enhancing student models in game-based learning with facial expression recognition. *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, 192–201.

[29] Sawyer, R., Smith, A., Rowe, J., Azevedo, R. and Lester, J. 2017. Is more agency better? The impact of student agency on game-based learning. *Proceedings of the 18th Int. Conference on Artificial Intelligence in Education*, 335–346.

[30] Snow, E., Likens, A., Jackson, G. and McNamara, D. 2013. Students' Walk through tutoring: Using a random walk analysis to profile students. *Proceedings of the 6th Int. Conference on Educational Data Mining*, 276–279.

[31] Snow, E., Likens, A., Allen, L. and McNamara, D. 2016. Taking control: Stealth assessment of deterministic behaviors within a game-based system. *Int. Journal of Artificial Intelligence in Education*. 26, 4, 1011–1032.

[32] Tóth, K., Greiff, S., Kalergi, C. and Wüstenberg, S. 2014. Discovering students' complex problem solving strategies in educational assessment. *Proceedings of the 7th Int. Conference on Educational Data Mining*, 225–228.

Identifying Profiles of Collaborative Problem Solvers in an Online Electronics Environment

Jessica Andrews-Todd
Educational Testing Service
660 Rosedale Rd.
Princeton, NJ 08540
+1(609) 734-5809
jandrewstodd@ets.org

Carol Forsyth
Educational Testing Service
90 New Montgomery, Ste. 1500
San Francisco, CA 94105
+1(415) 645-8465
cforsyth@ets.org

Jonathan Steinberg
Educational Testing Service
660 Rosedale Rd.
Princeton, NJ 08540
+1(609) 734-5324
jsteinberg@ets.org

André Rupp
Educational Testing Service
660 Rosedale Rd.
Princeton, NJ 08540
+1(609) 252-8545
arupp@ets.org

ABSTRACT

In this paper, we describe a theoretically-grounded data mining approach to identify types of collaborative problem solvers based on students' interactions with an online simulation-based task about electronics concepts. In our approach, we developed an ontology to identify the theoretically-grounded features of collaborative problem solving (CPS). After interaction with the task, students' log files were tagged for the presence of 11 CPS skills from the ontology. The frequencies of the skills were clustered to identify four unique profiles of collaborative problem solvers – Chatty Doers, Social Loafers, Group Organizers, and Active Collaborators. Relationships among cluster membership, task performance, and external ratings of collaboration provide initial validity evidence that these are meaningful profiles of collaborative problem solvers.

Keywords

Collaborative Problem Solving, Ontology, Assessment, Simulation-based Assessment, Discourse

1. INTRODUCTION

In our modern society, the nature of workplace performance has changed fundamentally through technology. An increasing number of complex tasks are being carried out in groups, often supported through digital tools with features that support collaboration. Accordingly, there has been increased attention in the assessment community on relevant competencies such as collaborative problem solving (CPS), a skill with multiple components that have been identified as important for success in the 21st century workforce [3].

Competency in CPS has been defined as "the capacity of an individual to effectively engage in a process whereby two or more agents attempt to solve a problem by sharing the understanding and effort required to come to a solution and pooling their knowledge, skills, and efforts to reach that solution" [17]. The complexity of this construct in having a cognitive dimension associated with problem solving processes and an interpersonal dimension associated with collaboration processes has made assessing CPS difficult, if not impossible, to carry out with traditional types of assessment such as multiple-choice questions with almost any sense of fidelity and generalizability [5]. As a result, there has been a turn to online learning environments such as games and simulations, which allow individuals to interact around complex problems and capture all actions and discourse in the environment as evidence of competency for assessment purposes.

While online environments offer promise for CPS assessment, there are challenges that exist. First, as with more traditional forms of assessment, assessment developers must conceptualize what skills define the construct and what actions and discourse would be indicative of those skills in the environment. Second, one must develop methods to make sense of the large streams of fine-grained data generated during real-time interaction in the environment [10].

In the current paper, we use a theoretically-grounded data mining approach [6] to discover profiles of various types of collaborative problem solvers that are strongly rooted in theory associated with collaboration, cognitive and social psychological research. Specifically, we describe the principled approach we used to conceptualize what skills make up the CPS construct, how we extracted evidence of those skills from the large streams of log data, and how we aggregated that information to create profiles that describe different types of collaborative problem solvers.

2. METHODS

2.1 Participants

Students in electronics and engineering programs were recruited from universities and community colleges across the United States. There were 129 individuals who completed the study in groups of three (i.e., 43 groups) that were randomly assembled. Of those students who reported their gender, 81% were males and 17% were females with 2% unreported. Of those who reported their race, 51% were White, 7% were Black or African American, 6% were Asian, 2% were American Indian or Alaska Native, 10% reported being more than one race, 2% reported Other, with 2% unreported. For ethnicity, 22% reported being Hispanic. The average age among students was 24 in a range of 16 to 60.

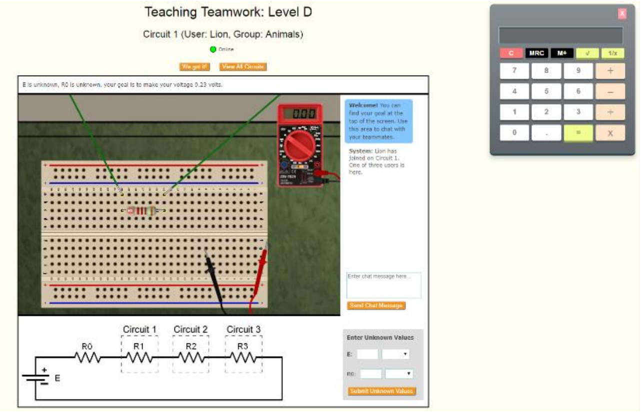
2.2 Task and Measures

Students completed a pre-survey that asked for their background information (e.g., age, gender, level of education) as well as their preferences for working in groups relative to independently and beliefs about the importance of collaboration. Instructors were then asked to randomly assemble their students into groups to complete an online simulation-based task on electronics concepts. The students worked in a computer lab and collaborated completely online in a computer-mediated environment described next.

In the task, called the Three-Resistor Activity, students worked in groups of three, each on a separate computer, and each running a fully functional simulation of a portion of an electronic circuit. The individual simulations were linked together to form a complete series circuit. The environment included a digital multimeter (DMM), two probes (red and black) from the DMM, a resistor, a calculator, a zoom button, a chat window, and a submit button (see Figure 1 for a screenshot of the task interface). These components allowed students to take measurements, view their circuit's resistance, perform calculations, zoom out to view (but not interact with) other teammates' circuits, communicate with teammates, and submit their work.

The individuals in each team were given the same task goal, which consisted of setting their resistors so that the voltage across these matched specified goal values. Since the circuits were connected in series, a change made to any one of these affected the current through the circuits and therefore the voltage drop across each of the circuits. Thus, rather than attempting to achieve the goal independently, team members needed to share information and coordinate their efforts to reach the goal voltage values across all the circuits. There were four levels of the task that increased in difficulty. At higher difficulty levels of the task, in addition to achieving their goal voltage values, the students were also asked to collaborate to determine the unknown resistance and supply voltage of an external, fourth circuit in the series. Students were allowed to communicate only using a chat window and could "zoom out" to see one another's circuits, but could only alter or make measurements on their own circuits. As students worked to achieve the goal voltages across four task levels, all of their relevant actions (e.g., DMM measurements, resistor changes, calculator entries, chat submissions) were time-stamped and logged to a database.

Table 1 provides an overview of the characteristics of each task level. Across the four task levels, the difficulty of the task increased either by presenting a more complicated problem (e.g., providing different goal voltages for each teammate in Level 2) or reducing the amount of information given (e.g., the external voltage in Levels 3 and 4). These changes increased the need for collaboration, as students were required to share more information and communicate more to identify unknown variables. Specifically, in Level 1, students were given the unknown resistance and supply voltage of an external, fourth circuit in the series and the goal voltages that needed to be reached were the same for each teammate. Having the same goal voltages for each circuit limited the amount of information that needed to be shared for each teammate to reach their goal. In Level 2, students were again given the unknown resistance and supply voltage of an external, fourth circuit in the series, but each teammate was now given a different goal voltage that they were required to reach. In Level 3, students were given the value of the resistance of the external circuit and again had different goal voltages to reach; however, the supply voltage of the external circuit was not provided. Thus, the team needed to reach the goal voltage for each circuit, but also discover and submit the supply voltage value and unit for the external circuit. In Level 4, students needed to discover and report the values and units for both the unknown resistance and the supply voltage of the external, fourth circuit as well as reach the specified and different goal voltages on each teammate's circuit.



Teaching Teamwork: Level D

Circuit 1 (User: Lion, Group: Animals)

E is unknown, R0 is unknown, your goal is to make your voltage 0.23 volts.

Welcome! You can find your goal at the top of the screen. Use this area to chat with your teammates.

System: Lion has joined on Circuit 1. One of three users is here.

Figure 1. Screenshot of the Three-Resistor Activity.

Table 1. Overview of Task Levels

Task Level	External Voltage (E)	External Resistance (R0)	Goal Voltages
1	Known by all teammates	Known by all teammates	Same for all teammates
2	Known by all teammates	Known by all teammates	Different for each teammate
3	Unknown by teammates	Known by all teammates	Different for each teammate
4	Unknown by teammates	Unknown by teammates	Different for each teammate

2.3 Competency Model

A CPS ontology (similar to a concept map) was developed to conceptualize the CPS construct. It provides a theory-driven representation of the targeted skills and their relationships, linking the skills to observable behaviors in the electronics task that would provide evidence of each skill. The top level of the ontology provides generalizable construct definitions for CPS (e.g., sharing information as one skill associated with the construct) that can be implemented in other work seeking to assess CPS or other related constructs. This top layer was developed based on an extensive literature review of CPS frameworks and other related research areas such as computer-supported collaborative learning, organizational psychology, individual problem solving, and linguistics [9, 12, 14, 15, 16, 17, 18, 22]. Each lower layer of the ontology becomes more specific describing CPS as interpreted within a domain (e.g., sharing status updates) and then within the task environment in the domain (e.g., sharing the status of the resistance in a circuit). Links between the layers describe how behaviors at lower levels can be combined to make inferences about cognitive behaviors at higher levels. In our research, the ontology designated the lower level features corresponding to over-arching social and cognitive dimensions. These lower level features were then extracted from log files prior to analysis. Figure 2 shows the structure for a portion of the CPS ontology with nodes corresponding to high-level CPS skills, sub-skills, features, and observable variables that can be inferred from the features, along with links indicating the relationships between the nodes.

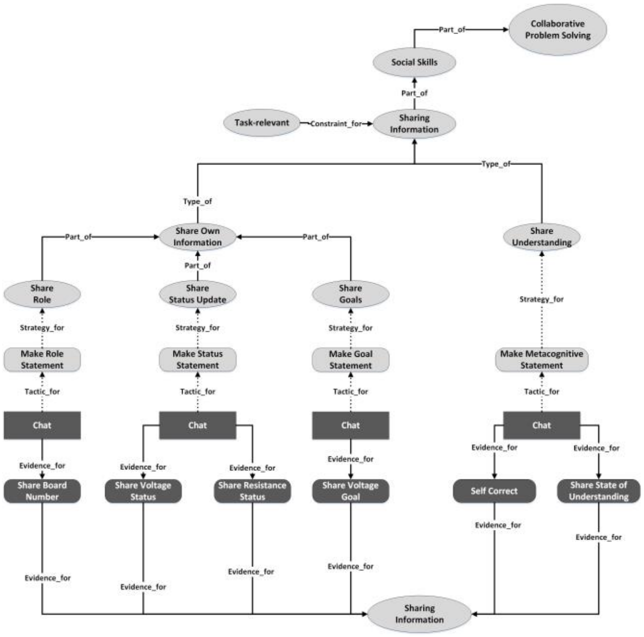


Figure 2. CPS ontology fragment structure.

The full ontology has nine high-level skills associated with CPS that we sought to identify in the data. Four skills correspond to the social dimension of CPS (i.e., maintaining communication, sharing information, establishing shared understanding, negotiating) and five skills correspond to the cognitive dimension of CPS (i.e., exploring and understanding, representing and formulating, planning, executing, monitoring). Maintaining communication corresponds to content irrelevant social communications [12]. This includes general off-topic communication (e.g., discussing what was eaten for breakfast), rapport building communication (e.g., greeting or praising teammates), and inappropriate communication (e.g., cursing). Sharing information corresponds to content relevant information communicated during collaboration. This includes the sharing of one's own information (e.g., sharing information related to the status of one's own work during the task), sharing task or resource information (e.g., communicating what tools are available in the task environment), and sharing understanding (e.g., sharing metacognitive information about the state of one's understanding). Establishing shared understanding corresponds to communicators attempting to learn the perspectives of others as well as trying to establish that what has been said is understood [4, 17]. This skill would include requesting information from teammates to verify that everyone has a common understanding, providing responses to teammates that verify comprehension of another's contribution, and making repairs when problems in shared understanding arise. Negotiating refers to communication that identifies whether or not conflicts exist in the ideas among teammates and seeks to resolve those conflicts when they arise [9]. This skill includes expressing both agreement and disagreement, and attempting to reach a compromise.

For the cognitive dimension, exploring and understanding refers to actions taken to build a mental representation of pieces of information associated with the problem. This includes interacting with the task environment to explore the problem space and demonstrating understanding of given information and information acquired while interacting with the environment. Representing and formulating refers to actions and communication in the service of building a coherent mental representation of the whole problem space. This includes developing a verbal or graphical representation of the problem and formulating hypotheses [17]. Planning corresponds to communication around developing a plan or strategy to solve the problem. This includes determining the overall goal, setting sub-goals or steps to carry out, and developing and revising strategies [9, 17]. Executing corresponds to actions and communication used in the service of carrying out a plan. This includes taking actions to enact a strategy, making suggestions for actions a teammate should carry out, and communicating to teammates the actions one is taking to carry out the plan. Monitoring refers to actions and communication associated with monitoring progress toward the goal and monitoring the organization of the team [16, 17]. This includes communicating one's own progress toward the goal, checking on the progress of teammates, and determining whether teammates are present and following the rules of engagement or their roles in completing tasks.

2.4 Qualitative Coding

The CPS ontology was used to create a rubric for raters to carry out qualitative coding of the log data to identify evidence of high-level CPS skills from low-level student discourse and actions. The nodes and links corresponding to each CPS skill in the ontology were transformed into extensive written protocols that included the high-level CPS skills, any sub-skills associated with the high-level skills, definitions for skills and sub-skills, example behaviors from the log data that would be indicative of each skill, and the action types associated with each skill (e.g., chat, calculation, measurement, submit). Two raters coded the content of students' discourse and their actions for the display of nine CPS skills. Evidence for two of the nine high-level CPS skills from the ontology could be found in both chats and actions (i.e., monitoring and executing) and were thus split into separate action and chat skills. As a result, the 11 coded skills were maintaining communication, sharing information, establishing shared understanding, negotiating, exploring and understanding, representing and formulating, planning, executing actions, executing chats, monitoring actions, and monitoring chats. Coding was done at the level of each log file event (i.e., each action submission or submission of a chat [utterance level] even if sequences of utterances mapped onto a singular CPS skill). Each of the 20,947 log file events only received one code. The inter-rater reliability between the two raters was high (Kappa = .84) based on a randomly selected sample of 20 percent of the data (approximately 4,200 events) that were double-coded.

On the social dimension, for maintaining communication, raters examined the log data for evidence of off-topic communication (e.g., "I should have drank coffee this morning"), rapport building communication (e.g., using chat emoticons, greeting teammates, apologizing, praising teammates), and inappropriate communication such as curse words or messages that degrade teammates (e.g., "you're an idiot"). For sharing information, raters looked for evidence of individuals sharing their own information for the problem (e.g., sharing what circuit board they were on, their goal voltage values, or resistance values on their board), sharing task or resource information (e.g., sharing where the zoom button was located, sharing that there was a calculator to use in the environment), and sharing their understanding (e.g., metacognitive statements such as "I don't get it"). For establishing shared understanding, raters looked for evidence of individuals requesting information from their partners (e.g., "what is your resistance?"

“what values do we need?”), and providing responses that indicate comprehension or lack of comprehension of a teammate’s statement (e.g., “ok,” “I hear you,” or requests for clarification). For negotiating, raters looked for evidence of individuals expressing agreement (e.g., “You are right”), expressing disagreement (e.g., “that’s not right”), and revising their own ideas or proposing alternate ideas.

On the cognitive side, raters looked for evidence of exploring and understanding by identifying actions in which individuals unsystematically made changes to task components in an effort to explore the interface. Unsystematic actions were defined as seemingly exploratory actions that were taken prior to developing a plan (e.g., spinning the dial on the digital multimeter, changing the resistance values several times in a few seconds). For representing and formulating, raters looked for evidence of individuals verbally communicating what the problem was (e.g., “this is a series circuit”) and communicating hypotheses for how their actions would affect the environment. For planning, raters looked for evidence of individuals communicating goals (e.g., “We need 6.69 volts across our resistors”) and communicating strategies to their teammates (e.g., “ok we set our values to R and find current”). For executing actions, raters looked for actions that individuals took to carry out the plan or strategy (e.g., changing their voltage values to the voltage suggested by a teammate or performing a calculation associated with Ohm’s Law). For executing chats, raters looked for evidence of individuals making suggestions or directing their teammates to perform actions associated with their plan (e.g., “Adjust yours to 300 ohms”) and reporting their own actions that they were taking to carry out the plan (e.g., “Let me go a little lower and then readjust”). For monitoring actions, raters looked for evidence of individuals carrying out actions associated with monitoring the team’s progress toward the goal (e.g., clicking the submit button to receive feedback about success in solving the problem) or monitoring teammates (e.g., using the zoom feature to view the state of a teammate’s circuit board). For monitoring chats, raters looked for evidence of individuals stating the result of their monitoring of progress toward the goal (e.g., “I’ve got my goal voltage”), monitoring the status of teammates (e.g., “Where is Rain?”), and prompting teammates to perform tasks (e.g., “Let’s get a move on Sleet”).

3. ANALYSES AND RESULTS

The analyses were conducted in two stages. First, the frequencies of the 11 CPS skills displayed by each individual were clustered with a hierarchical approach to discover meaningful profiles. Second, the profiles were validated by their relationship to performance and self-report measures with non-parametric inferential statistical tests and Monte Carlo simulations due to the abnormal distributions of the variables.

3.1 Cluster Analysis and Profiles

We chose an exploratory clustering method [21] for uncovering potential profiles of collaborative problem solvers in part because we had no formal a priori theory regarding the number and composition of these profiles. Additionally, as the sample size (N=129) did not warrant methods like K-means which are typically applied to larger samples [13], Ward’s Method was employed to cluster the frequencies of each CPS skill displayed to allow us to examine the breakdown of possible clusters so that a meaningful number of clusters could be chosen. The final number of clusters was determined based on an initial interpretation of the theory stated in existing literature in collaboration and psychological research. Thus, these are preliminary findings and to date no gold standard exists for the collaborative problem solving domain.

A four-cluster solution was most defensible from a theoretical perspective and the expected relationships to other variables that resulted which will be explained in later sections; Table 2 shows the frequencies for this solution. Specifically, the learners in the four clusters differed systematically in the frequencies of CPS skills that were displayed. The four clusters were named Chatty Doers, Social Loafers, Group Organizers, and Active Collaborators. In the next section, we describe the key behavioral patterns in each cluster based on CPS skill frequencies standardized to the total sample and discuss the relevant theory explaining the type of collaborative problem solver that may display the patterns of behavior.

Table 2. Collaborative Problem Solver Profiles

Profile	Frequency	Percent of Sample
Chatty Doers	35	27.1
Social Loafers	68	52.7
Group Organizers	16	12.4
Active Collaborators	10	7.8

3.1.1 Chatty Doers

Students in Cluster 1, labeled “Chatty Doers” (n=35) were high ($z \geq 0.20$) on executing actions and maintaining communication, somewhat high ($0.10 \leq z < 0.20$) on planning and sharing information, and were low ($z \leq -0.20$) on monitoring actions. These students were labeled “Chatty Doers” due to their high levels of maintaining communication chats and executing actions. Chats associated with maintaining communication were communications that were social in nature, but not relevant to solving the problem [12]. These included discussing what one did last week, discussing homework from the night before, and praising teammates. Thus, these individuals were designated as chatty more generally given their off-topic, social communication that was absent of high levels of communication related to skills such as negotiating or establishing shared understanding. These individuals also engaged in a high level of executing actions relative to other individuals which included making resistor changes and performing calculations. Thus, these individuals were the doers carrying out many of the actions associated with executing the team’s plan.

3.1.2 Social Loafers

The standardized means for Cluster 2, labeled “Social Loafers” (n=68) displayed below average demonstration ($z < 0.00$) of almost all skills. These students were named “Social Loafers” given their low levels of the CPS skills which may be explained by a social psychological phenomenon in which individuals decrease their individual effort when working in groups [11] as they each assume another member will take the lead in solving the problem. Students in this cluster appeared to do just this as they engaged in fewer collaborative problem solving behaviors relative to other individuals.

3.1.3 Group Organizers

The standardized means for Cluster 3, labeled “Group Organizers” (n=16) showed high demonstration ($z \geq 0.20$) of monitoring actions, representing and formulating, and negotiating, somewhat high demonstration ($0.10 \leq z < 0.20$) of executing chats and sharing information, and low demonstration ($z \leq -0.20$) of planning. These students were named “Group Organizers” due to their high levels of communications and actions associated with establishing and maintaining organization for the problem and the group [17]. This included things such as monitoring behaviors like using the zoom feature to monitor the state of teammates' behaviors and circuit boards, verbally representing the problem for teammates, and communicating important information to group members such as what actions are being taken to solve the problem, all of which can be in the service of keeping the group organized.

3.1.4 Active Collaborators

The students in Cluster 4, referred to as the "Active Collaborators" (n=10) showed above average demonstration (z > 0.00) of almost all skills, though they demonstrated low levels ($z \leq -0.20$) of maintaining communication. Cluster 4 students were named "Active Collaborators" given their high levels of almost all of the social and cognitive processes associated with CPS [8].

3.2 CPS Skill Profile Validation

The CPS skill profiles were validated by relating the cluster membership assignment to performance metrics from the task and scores from student self-reports of preference in working with others. Prior empirical studies suggest a positive relationship between demonstration of collaborative behaviors and performance outcomes [1, 8], thus we hypothesized that students demonstrating more of the skills associated with CPS would have greater success on the task as measured by the number of levels completed in the task. Number of task levels completed was treated as an individual performance measure, though contributions of other teammates could impact the score. In regard to self-report measures, we were unsure as to whether students would accurately report whether or not they thought they were good collaborators but suspected they would answer more honestly as to whether or not they preferred to work alone, thus the latter question was asked to students along with their perceptions of how important collaboration is in the real world. The cluster membership assignment, the performance metrics, and the self-ratings were submitted to Kruskal-Wallis tests with a Monte Carlo simulation to determine the significance of the relationships among the variables.

3.2.1 Cluster Membership and Performance

There was a significant relationship between cluster membership and success on the task levels (i.e., number of task levels completed) ($X^2(3,126) = 6.93$, p <.05 with a one-tailed test, *partial* η^2 =.053). The Monte Carlo simulation with 10,000 test samples revealed a p value of .032 (lower bound = .023; upper bound = .036). The mean ranks of the different groups based on completed task levels showed patterns in line with our prediction. Specifically, the Active Collaborators had the highest mean rank of 93.95 whereas the Social Loafers had the lowest mean rank of 61.65. Chatty Doers and Group Organizers fell in between these two groups with mean ranks of 63.89 and 63.59, respectively. Post hoc comparisons with a Bonferroni correction revealed that there was a significant difference between the Social Loafers and Active Collaborators (p = .027) and a marginally significant difference between the Chatty Doers and Active Collaborators (p = .063) in terms of mean rank of performance. All other comparisons were not significant. These results make sense as we would expect the Active Collaborators to be the high performers given that they demonstrated high frequencies of all of the necessary attributes that we had identified for effective collaborative problem solvers. It also makes sense that Social Loafers performed the poorest as these individuals demonstrated lower incidences of CPS skills.

After confirming that there was indeed a significant difference in the relationship between performance and type of collaborative problem solver, we moved on to compare cluster membership to self-reported collaboration preferences.

3.2.2 Cluster Membership and Collaboration Preferences

Recall that students completed a pre-survey that included questions about their preferences in working with others and how much they valued collaboration in the real world. We explored how responses to these questions were related to cluster membership. There was a marginally significant relationship between cluster membership and response to the question about whether or not students preferred to work alone (X^2 (3,126) = 7.23, p = .065 with a two-tailed test, *partial* η^2 =.055). The Monte Carlo simulation revealed a p value of .064 (lower bound = .057; upper bound = .070). The mean ranks for responses - where higher numbers indicate stronger preference to work alone - were as follows: Social Loafers (71.05), Chatty Doers (54.90), Group Organizers (54.38), and Active Collaborators (47.10). The direction of these results are consistent with what would be expected. Social Loafers who demonstrate few CPS skills and seem to expend little effort during collaborative activity would be expected to prefer to work alone. Conversely, Active Collaborators who demonstrate high incidences of CPS skills and are thus active during collaborative activity would be expected to have a preference to work with others. Chatty Doers and Group Organizers who display CPS skills, but not to the extent of Active Collaborators would be expected to fall in between the Active Collaborators and Social Loafers.

The students were also asked about their ratings as to how important collaboration is to the real world. Cluster membership had a non-significant relationship to responses on this question (p = .465). The mean ranks where higher numbers indicate higher importance for collaboration in the real world were as follows: Group Organizers (71.94), Chatty Doers (68.82), Active Collaborators (62.90), and Social Loafers (59.82). One possible explanation for this finding is that instructors likely informed students about the importance of collaboration in setting up the study activity so student responses may have been influenced by this information. The mean ranks were relatively high for all groups so this explanation may be appropriate, but further testing is necessary to draw any strong conclusions.

4. CONCLUSIONS

Many methods exist for discovering profiles of how students collaborate during problem solving (for a review see [7]). In the current study, we used a frequency-based cluster approach to discover cluster profiles, following a previously established approach [8]. This approach was chosen because we are discovering profiles of types of collaborative problem solvers in a discovery learning environment. That said, we acknowledge that other approaches could be considered, though they may not be the best fit in the given context. For example, for an analysis of CPS in an international assessment context [17], students interacted with a constrained environment (e.g., a dropdown menu for chat choices) making it possible for traditional psychometric approaches to sufficiently analyze the student responses and communication. Conversely, in previous research on serious games with collaboration, an Epistemic Network Analysis (ENA) approach has been used to analyze how students connect knowledge and skills during collaboration over time [19]. However, the focus of our investigation is on collaboration without including domain knowledge, though we plan on augmenting the ENA approach for our purposes in future analysis. Additional approaches focusing on group dynamics [e.g., 20] were not chosen as the goal of this investigation was to analyze student collaboration on an individual level. Therefore, we are not stating that our educational data mining approach is the only means to analyze CPS skills, but rather that it may be most appropriate for profiling individual students for CPS skills without including domain knowledge or group dynamics.

In our implementation of the frequency-based cluster approach, we demonstrated that meaningful results can emerge from incorporating theory into the approach to identify types of collaborative problem solvers. Specifically, the current approach yielded four types, namely, Chatty Doers, Social Loafers, Active Collaborators, and Group Organizers in our assessment context. The Chatty Doers displayed high levels of maintaining communication chats, or content irrelevant, social communication, and high levels of executing actions in the service of solving the problem. The Social Loafers were characterized by low levels of CPS skills in general whereas Active Collaborators were characterized by high levels of all CPS skills except maintaining communication. Group Organizers were categorized by CPS skills associated with establishing and maintaining organization for the problem and the group. Over half of the students demonstrated behaviors characteristic of Social Loafers while few students were characterized as Active Collaborators.

The profiles showed expected relationships with performance. Specifically, the Active Collaborators showed the highest levels of performance whereas the Social Loafers showed the lowest levels of performance. The performance of Chatty Doers and Group Organizers fell in between these groups. These results are consistent with prior work showing positive social and cognitive behaviors benefiting performance outcomes [8] and non-collaborative behaviors hurting performance outcomes [2]. The four cluster profiles also showed a marginally significant relationship with a self-report measure of whether or not students preferred to work with others. Social Loafers had the highest ratings of preferring to work alone perhaps because these students are less willing to expend the effort needed to sustain collaborative relationships to work with others as compared to their peers. Conversely, the Active Collaborators preferred to work with others more than did other students. This makes sense as these students are active during collaboration and thus likely willing to expend the effort needed to work with others to solve problems.

Perhaps the most important feature of this study is not necessarily the profiles themselves but rather the blending of theory with educational data mining techniques. All features of CPS were defined a priori based on a theoretically-grounded ontology with multiple levels and two dimensions of social and cognitive skills. In total, this ontology defines nearly 51 features. This method may be helpful in discovering meaningful relationships between variables in large log files from games and simulations. Furthermore, the number of clusters was defined based on theoretical grounding. We deemed the method successful based on the meaningful profiles discovered and preliminary relationships to external measures, all of which can be explained by psychological research. In the current paper, we coded high-level CPS skills based on low-level student behaviors. In future work, we intend to code at a lower, sub-skill level and incorporate methods to aggregate to higher levels in the ontology. Due to the time-intensive nature of human coding with these kind of data, we further plan to explore the possibility of automating the coding of chat data using machine learning algorithms.

There are some limitations to this study. One involves the small number of participants compared to the number of CPS skills we were attempting to measure. Additionally, we had few items to use as external correlates to our cluster profiles. In follow-up research, we are currently conducting a study with a larger sample to confirm the existence of the profiles discovered in this study and administering multiple well-constructed external measures that can potentially help build a validation argument for any discovered profiles. Another limitation of this study is that the measure used for performance outcomes incorporated the contributions of group members. As we are investigating CPS on an individual level, it would be ideal to compare student skills on an individual level to a performance measure for each individual. Thus, in an upcoming study, we have also incorporated a measure of performance that may more closely resemble individual performance but complete exclusions of group dynamics is difficult in the given environment. Thus, follow-up analyses on the group dynamics and composition are currently underway.

The current study provides preliminary results that will greatly inform the work on the upcoming data collection. Furthermore, the current study views collaboration through the lens of the Three-Resistor Activity; however, our intention is to draw upon a wide variety of tasks and content areas in upcoming studies. This future work will allow us to explore the generalizability of the CPS ontology, as its structure allows for decoupling it from content and modifying lower-level nodes to support features in other tasks.

Overall, the study demonstrates a methodology that incorporates well-detailed theory and measures emerging from the learning sciences and blends it with educational data mining. This approach resulted in meaningful profiles constructed from features defined a priori, and can serve as an example for how to combine theory and data-driven approaches to make meaningful inferences about students' knowledge, skills, and abilities from interactions in an online environment.

5. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant DUE 1535224 awarded to the first author. The opinions expressed are those of the authors and do not necessarily represent views of the National Science Foundation.

6. REFERENCES

[1] Andrews, J. J. and Rapp, D. N. 2015. Benefits, costs, and challenges of collaboration for learning and memory. *Translational Issues in Psychological Science*, 1, 2, 182-191.

[2] Andrews, J. J., Kerr, D., Mislevy, R. J., von Davier, A. A., Hao, J., & Liu, L. (2017). Modeling collaborative interaction patterns in a simulation-based task. *Journal of Educational Measurement, 54*(1), 54–69.

[3] Burrus, J., Jackson, T., Xi, N., and Steinberg, J. 2013. *Identifying the most important 21st century workforce competencies: An analysis of the occupational Information network (O* NET).* ETS Research Report RR-13-21. Educational Testing Service, Princeton, NJ.

[4] Clark, H. H. 1996. *Using language*. Cambridge University Press, New York, NY.

[5] Davey, T., Ferrara, S., Shavelson, R., Holland, P., Webb, N., and Wise, L. 2015. *Psychometric considerations for the next generation of performance assessment*. Educational Testing Service, Princeton, NJ.

[6] Forsyth, C.M., Graesser, A. C., Pavlik, P., Millis, K., and Samei, B. 2014. Discovering theoretically grounded predictors of shallow vs. deep- level learning. In Stamper, J.,

Pardos, Z., Mavrikis, M., McLaren, B.M., Eds. In *Proceedings of the 7th International Conference on Educational Data Mining (EDM 2014).* International Educational Data Mining Society, 229-232.

[7] Graesser, A.C., Cai, Z., Hu, X., Foltz, P.W., Greiff, S, Kuo, B-C. Liao, C-H. , and Shaffer, D.W. 2017. Assessment of collaborative problem solving. In R. Sottilare, A. Graesser, X. Hu, & G. Goodwin, Eds. *Design Recommendations for Intelligent Tutoring Systems: Volume 5 – Assessment,* U.S. Army Research Laboratory, Orlando, FL, 275-285.

[8] Herborn, K., Mustafić, M., and Greiff, S. 2017. Mapping an experiment-based assessment of collaborative behavior onto collaborative problem solving in PISA 2015: A cluster analysis approach for collaborator profiles. *Journal of Educational Measurement,* 54, 1, 103–122.

[9] Hesse, F., Care, E., Buder, J., Sassenberg, K., and Griffin, P. 2015. A framework for teachable collaborative problem solving skills. In P. Griffin and E. Care, Eds. *Assessment and teaching of 21st century skills.* Springer, New York, NY, 37–56.

[10] Kerr, D., Andrews, J. J., and Mislevy, R. J. 2016. The in-task assessment framework for behavioral data. In A. A. Rupp and J. P. Leighton, Eds. *The handbook of cognition and assessment: Frameworks, methodologies, and applications*. Wiley-Blackwell, Hoboken, NJ, 472-507.

[11] Latané, B., Williams, K., and Harkins, S. 1979. Many hands make light the work: The causes and consequences of social loafing. *Journal of Personality and Social Psychology,* 37, 6, 822-832.

[12] Liu, L., von Davier, A. A., Hao, J., Kyllonen, P., and Zapata-Rivera, J.-D. 2015. A tough nut to crack: Measuring collaborative problem solving. In Y. Rosen, S. Ferrara, and M. Mosharraf, Eds. *Handbook of research on computational tools for real-world skill development*, IGI-Global, Hershey, PA, 344–359.

[13] MacQueen, J. B. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability.* University of California Press, Berkeley, CA, 281-297.

[14] Meier, A., Spada, H., and Rummel, N. 2007. A rating scheme for assessing the quality of computer-supported collaboration processes. *International Journal of Computer-Supported Collaborative Learning,* 2, 1, 63–86.

[15] Morgan, B. B., Salas, E., and Glickman, A. S. 1993. An analysis of team evolution and maturation. *Journal of General Psychology*, 120, 3, 277–291.

[16] OECD. 2013a. *PISA 2012 assessment and analytical framework: Mathematics, reading, science, problem solving and financial literacy*. OECD Publishing, Paris.

[17] OECD. 2013b. *PISA 2015 collaborative problem solving framework*. OECD Publishing, Paris.

[18] O'Neil, H. F., Chuang, S., and Chung, G. K. W. K. 2003. Issues in the computer-based assessment of collaborative problem solving. Assessment in Education: *Principles, Policy & Practice,* 10, 3, 361–373. DOI: https://doi.org/10.1080/0969594032000148190

[19] Shaffer, D. W., Hatfield, D., Svarovsky, G., Nash, P., Nulty, A., Bagley, E. A., Frank, K., Rupp, A.A., and Mislevy, R. J. 2009. Epistemic Network Analysis: A prototype for 21st century assessment of learning. *The International Journal of Learning and Media,* 1, 1, 1–21.

[20] Von Davier, A. and Halpin, P. 2013. *Collaborative problem solving and the assessment of cognitive skills: Psychometric considerations.* ETS Research Report RR-13-41. Educational Testing Service, Princeton, NJ, 1-42.

[21] Ward, Jr., J. H. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association,* 58, 301, 236-244. DOI: 10.2307/2282967

[22] Zhuang, X., MacCann, C., Wang, L., Liu, L., and Roberts, R. D. 2008. *Development and validity evidence supporting a teamwork and collaboration assessment for high school students.* ETS Research Report RR-08-50. Educational Testing Service, Princeton, NJ.

Data-Driven Approach Towards a Personalized Curriculum

Michael Backenköhler
Saarland University

Felix Scherzinger
Saarland University

Adish Singla
MPI-SWS

Verena Wolf
Saarland University

ABSTRACT

Course selection can be a daunting task, especially for first-year students. Sub-optimal selection can lead to bad performance of students and increase the dropout rate. Given the availability of historic data about student performances, it is possible to aid students in the selection of appropriate courses. Here, we propose a method to compose a personalized curriculum for a given student. We develop a modular approach that combines a context-aware grade prediction with statistical information on the useful temporal ordering of courses. This allows for meaningful course recommendations, both for fresh and senior students. We demonstrate the approach using the data of the computer science Bachelor students at Saarland University.

1. INTRODUCTION

Students at higher education institutions usually have to choose from a large set of possible courses in order to achieve an academic degree. Even for senior students, it is not obvious which courses to follow and in what sequence as the number of possible choices is large. Students often have problems to ensure progress in a program of study, especially in the first years of study, and to graduate in a timely manner.

Student success is also an important objective for decision makers at universities, which continuously monitor dropout rates and average times to degree. Completion rates at European universities range between 39% to 85% and are highly program dependent, while the average time-to-degree is around 3.5 years for a Bachelor degree [17].

When pursuing a degree students typically have to complete a set of mandatory courses, as well as courses that can be chosen more freely. In the first years, an adequate order of mandatory courses is of interest while in later years the focus is on the question which courses to take in general and which not. Instead of relying on individual recommendations from other students, our goal is to take advantage of the combined experience of former students and address both, an adequate temporal ordering and an intelligent selection of courses.

We propose an approach that combines statistical methods based on course orderings and grade prediction based on a collaborative filtering approach. This results in a model consisting of two main components, a course dependency graph and grade prediction. Therefore our model combines two major criteria: The expected performance, i.e. the expected grade, and preparedness, i.e. how prior course choices may benefit the student, for a given course. We believe that weaving the two criteria strongly increases the usability of our recommendations compared to previous work focusing only on one of the two.

To train our model we use long-term educational data of computer science Bachelor students from Saarland University's computer science department. The data consists of course performance information from several thousand students of various countries during the last ten years. Experiments with a first subset of students already showed promising results giving recommendations for first-year as well as for senior students.

2. RELATED WORK

Many course recommendation approaches are based on *performance prediction.* A wide range of standard machine learning methods have been applied to this problem [14, 15], as well as recommender system techniques [10]. Ray and Sharma [8] apply collaborative filtering based on item-item similarity. Ren et al. [9] supplement a matrix factorization approach with weights for recently taken courses. Besides a gain in predictive quality, the resulting model carries valuable information on beneficial orderings of courses. Polyzou and Kyrapis [7] propose a matrix factorization based on course-specific features. Slim et al. [12] use Markov networks of courses to predict individual grades and estimate the future performances inside a study program.

In contrast to the aforementioned approaches, our technique separates the concerns of performance and preparedness. This has the benefit of allowing for a custom weighting of the two components, as well as the increased explanatory value of the model itself.

Much effort on curriculum planning has been focused on Massive Open Online Courses (MOOC). For instance, Hansen et al. [5] analyse characteristic question sequences in online courses by applying Markov chains to student clusters. Chen et al. [3] propose a squencing for items in the context of web-based courses.

In the context of university eduction, much effort has been directed towards providing analytical tools to educators and institutions. For example Zimmermann et al. [18] predict graduate performance, based on the students' undergraduate performances. Saarela and Kärkkäinen [11] analyse undergraduate student data to indentify relevant factors for a successful computer science education.

3. PROBLEM SETTING

We consider the problem of designing a student's curriculum that optimizes performance (measured in terms of course grades) and the time to degree. Hence, for each semester a subset of the courses offered is chosen such that the student's complete trace from the first semester until the final degree is (approximately) optimal, i.e., the performance and time to degree does not improve if the order in which the courses are taken is changed or if different courses are taken. We assume that a large number of traces of former students are given, including the particular grades achieved in each course. Note that this also includes data of students retaking courses are failing.However, the data may not provide information about students that enroll in a course but withdraw before the final exam. In addition, we assume that recommendations for students that already participated in certain courses, the corresponding partial trace is available as well as meta data about the student. Moreover, we want to take into account all selection rules of the corresponding study program.

The data-set consists of performance and meta-information of the students at the computer science department of Saarland University since 2006. It includes grades, basic information regarding students (age, nationality, sex, course of studies) as well as basic information regarding the lecture (course type, lecturer). Here, we consider a subset of 72 recurring courses which have a total of 16,090 entries of 1,700 students. A challenge regarding this particular data set is the fact that students may register fairly late in the semester for a particular course. Therefore the data does not capture a early student drop out.

4. COMPONENTS OF OUR APPROACH

In the context of standard recommender systems, the predicted rating is the basis for a recommendation. However, in the context of course recommendation, further aspects, such as the knowledge gain and constraints of the study program have to be taken into account. Here, we present an approach that is flexible enough to also incorporate such criteria in a modular way. Moreover, in our approach selection criteria can further be prioritized by the student. A student may, for example, prioritize taking a course that increases the preparedness for certain other courses. In this case, the course may be recommended although the students performance alone did not lead to suggestion of that course.

We construct a personalized *recommendation graph* of courses for each student based on the two main components: the course dependency graph and the performance prediction. The course dependency graph aims to capture the positive effect that course A has on the performance in course B. The performance prediction is done using a *collaborative filtering* approach, that incorporates contextual features of both the student and the course.

4.1 Course Dependency Graph

The Course Dependency Graph is a graph whose node set equals the set of all (regularly or irregularly offered) courses. A directed edge between course A and course B means that when passing A before B then the chance of getting a better grade in B is higher compared to the grade in B obtained for the order B before A.

We use the *Mann-Whitney U-test* [2] to construct such a graph of courses. The hypothesis of the test is that one random variable is smaller than another. If we let the random variable $X_{<c}$ denote the grade in course B for a student that had a grade $< c$ in course A an edge represents the hypothesis

$$\Pr(X_{<c} < k) > \Pr(X_{\geq c} < k),$$

where $X_{\geq c}$ includes the case of not taking course A. The hypothesis describes that the probability of drawing a grade of subset $X_{<c}$ which is better than k is higher than doing the same for subset $X_{\geq c}$.We fix a small significance level $\alpha = 0.0001$, to find the most important course relations. Since the test is quite sensitive, it tends to identify too many course pairs for higher significance levels. Moreover, a minimum number of 20 samples is required for each case to perform the test. The graph only contains an edge between two courses if the test confirms the above hypothesis.

In Germany grades are numbers in the set

$$P = \{1, 1.3, 1.7, 2, 2.3, 2.7, 3, 3.3, 3.7, 4, 5\},$$

where lower numbers are better and 5 is the failing grade. In general, we assume these performances to be normalized to mean zero and unit variance w.r.t. courses.

To construct the course dependency graph, we first construct one graph for each grade threshold $c \in P$. Next we average over the edges of all graphs, resulting in edge weights between 0 and 1. In this way the final graph, in which course dependency is not binary but a weighting, is more informative. A large value implies that this course ordering is beneficial to students of all performance levels while a low value indicates that this ordering is only helpful for a smaller set of students. Note that the absence of edges indicates that there is not enough information about the relation between the two courses.

An excerpt of a course dependency graph is shown in Figure 1. We find that 'Programming I', 'Maths I' and 'Maths II' are good starting points in this graph for a first-year student as they do not have incoming edges. Note that the missing edge between 'Maths I' and 'Maths II' is meaningful as 'Maths I' focuses on Linear Algebra while 'Maths II' is concerned with Analysis. As opposed to this, for 'Programming I' and 'II' the graph suggests to first take 'Programming I' as a preparation, which is a meaningful recommendation as the contents of 'Programming II' are based on those of 'Programming I'. Moreover, the graph shows a number of less obvious relations between courses (e.g. 'Programming II' and 'Theoretical Computer Science').

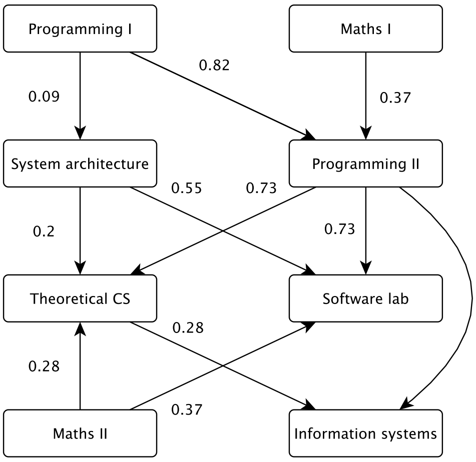


Figure 1: Excerpt of a course dependency graph, based on *Mann-Whitney U-test* with significance level of 0.0001, representing the dependencies between most of the basic courses in CS curriculum at Saarland University.

4.2 Grade prediction

We use a *collaborative filtering* [10] approach to predict student performance. One advantages of this approach is that no imputation of missing entries is necessary but the optimization only runs over existing entries.

We associate with each student i and course j an n-dimensional feature vector, s_i and c_j, respectively. The predicted performance is the cross-product of both vectors, i.e.

$$f(i,j) = \langle s_i, c_j \rangle = \sum_{k=1}^{n} s_{i,k} c_{j,k} \,,$$

which we call the *predictor* function. Let $g_{i,j}$ be the performance of a student i in course j and let $\mathcal{G}_t$ denote the set of all known performances of students up to semester t. Then the standard loss is the regularized MSE, i.e.

$$L(S,C,t) = \sum_{g_{i,j} \in \mathcal{G}_{t-1}} \left(f(i,j) - g_{i,j}\right)^2 + \lambda h(S,C)$$

with regularization term

$$h(S,C) = \sum_{i \in S} \|s_i\| + \sum_{j \in C} \|c_j\| \,,$$

where S is the set of all students and C the set of all courses.

4.2.1 Contextual Information

The above loss metric only depends on information about the students' performances, i.e. their grades. However, the *context* of a performance can contain vital information. Usually, in the context of student records a wealth of data is readily available. This includes meta-data of a student such as age, gender, and nationality and data regarding the progression of the student throughout study programs. Moreover, information regarding the course, such as the lecturer, is typically known.

A standard and straight-forward, approach to include such information is to *pre-filter* data [10]. This entails partitioning data along contextual criteria and then training a model for each subset. Here, the only performed pre-filtering is to take only computer science Bachelor students into account. Other partitionings, e.g. partitioning along the semester, have not improved predictive quality.

Further contextual information is included explicitly in the model as follows. The predictor f is augmented by linear terms for contextual features. Categorical features, such as teachers, are one-hot encoded. Continuous features are centered to zero mean and unit variance. In principle we can introduce these additional linear parameters for both, courses and students, but it turns out that the best results are achieved if we associate features with courses. Given the large number of contextual features it proved advantageous to set up a feature selection pipeline in which certain features are identified for each course. Specifically, features were identified by using a 5-fold cross-validated *recursive feature elimination*. Therein features are iteratively removed according to their coefficient in a linear model. The cross-validation is used to determine the number of features kept. Thus, the predictor becomes

$$\tilde{f}(i,j,t) = \langle s_i, c_j \rangle + \langle ctx(i,j,t), c_j^{ctx} \rangle \,,$$

where ctx is the performance context according to the above feature selection pipeline. Consequently, the parameter vector for course j becomes

$$\tilde{c}_j = (c_{j,1}, c_{j,2}, \ldots, c_{j,n}, c_{j,1}^{ctx}, \ldots, c_{j,m_j}^{ctx})$$

and m_j is the number of features selected for the context of a performance in course j.

Another key property to be considered when working with past performances is the temporal distance to the current time. A performance achieved one semester ago should be considered more important than one five semesters ago [9]. Therefore it is natural to add a temporal decay to the loss function. Considering the semester t' of a specific performance $g_{i,j,t'}$, we can multiply an exponential decay function. Thus, the now time-dependent loss is

$$L(S,C,t) = \sum_{g_{i,j,t'} \in \mathcal{G}_{t-1}} e^{-\alpha \cdot (t-t')} \left(\tilde{f}(i,j,t) - g_{i,j,t'}\right)^2 + \lambda h(S,C) \,, \quad (1)$$

where $\alpha > 0$ is the temporal decay parameter.

4.2.2 Minimization

The non-linear minimization problem in Eq. (1) is of high dimensionality because of the parameter vectors s_i and c_j for $i \in S, j \in C$. It can most effectively be achieved using stochastic gradient descent techniques with adaptive learning rates, because for this approach course vectors stabilize more quickly. Specifically, we used the Adagrad algorithm [4], which avoids strong alteration of frequently considered parameters, which is the case for many course parameters, while seldomly encountered parameters may be altered more, which is fitting for student parameters. We fixed a batch size of 1000 and performed 500,000 iterations of the algorithm. Each minimization is performed for 5 different initial random values. The value according to the smallest training loss is selected. This was performed for all semesters in a grid search over different dimensionality parameters and regularization parameters, i.e. for parameter tuples (λ, n). Before minimization the data was normalized along the lectures to zero mean and unit variance.

4.2.3 Evaluation

The most natural approach to evaluate the model is to split the data by semesters. Given a fixed semester t the data up to (including) semester $t-1$, i.e. $\mathcal{G}_{t-1}$, is used as a training set. The data of semester t, i.e. $\mathcal{G}_t \setminus \mathcal{G}_{t-1}$ is used as a test set.

The measures of quality we use are the mean absolute error (MAE) and the root mean square error (RMSE). As a baseline we provide the RMSE and MAE for the mean predictor with respect to both, the students and the courses in Table 1.

In the evaluation of the context-free model, we see, that low-dimensional models (i.e. models with only few features) perform best. The absolute values of these errors are further improved by pre-filtering the data considered. If, for example, only Bachelor computer science students are considered the test error decreases. The decay factor leads to an improvement. For example, for $n = 1$ and $\lambda = 0.1$ the MAE decreases from 0.856 to 0.852. In Figure 2 the prediction results for the importance decay $\alpha = 0.1$ are shown. Given this loss function, the one-dimensional, less regularized model outperforms the others in terms of both, the MAE and the RMSE. The inclusion of contextual information leads to a further reduction, such that for $n = 1$ and $\lambda = 0.1$ the MAE is 0.8459, while the RMSE is 1.0904.

Table 1: The RMSE and MAE for the mean predictors along the student and the course axis, respectively.

	MAE	RMSE
course	1.1130	1.3311
student	0.9268	1.1883

5. RECOMMENDATION SYNTHESIS

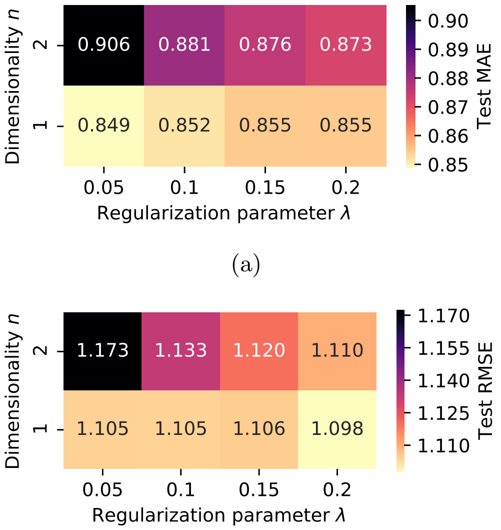
The recommendation combines the course dependency graph, the grade prediction, and constraints based on the study regulation in order to compute a *recommendation score*. A larger score corresponds to a stronger recommendation.

5.1 Combining the Components

The recommendation score for a course j w.r.t. a student i combines several criteria, namely the preparedness for j, the general merit of j, and the predicted performance of i in course j.

Let R_i denote the set of courses that student i has finished within the last t semesters. Now, for each course $j \in C \setminus R_i$, we sum over the weights of the edges of the course dependency graph that start in some course $j' \in R_i$ and end in j. This value is an approximation for the preparedness $p_{i,j} \in \mathbb{R}_{\geq 0}$ of the student w.r.t. course j.

For the general merit of a course, we use the out-degree of the course $\deg^+(j)$ in the graph as an approximation of its benefit towards other courses. Note that this criteria is especially relevant for first-year students as for them nodes with higher out-degree provide a good starting point. Further note that for such students, $R_i = \emptyset$ and the grade prediction can only give average values as no information about their previous performance is available.



(a)

(b)

Figure 2: The MAE (a) and RMSE (b) for different dimensionalities n and regularization parameters λ. The models were trained and tested on Bachelor CS students only. The loss is weighted by time with $\alpha = 0.1$.

To incorporate information about the predicted performance, we transform the predicted grades $\hat{g}_{i,j}$, such that good grades map to large values and poor grades to small values, i.e., we consider the value $(5 - \hat{g}_{i,j})/4 \in [0, 1]$.

We parameterize these factors into a linear model, that gives us a raw, unfiltered recommendation value

$$r'_{i,j} = c_p p_{i,j} + c_g (5 - \hat{g}_{i,j})/4 + c_m \deg^+(j), \qquad (2)$$

where $c_p, c_g, c_m \in [0, 1]$ provide a weighting for the three factors, i.e., $c_p + c_g + c_m = 1$.

We finally filter the recommendations as follows. The choice of courses is constrained by study regulations. Thus, for a given student i, some courses may not contribute towards completion of the program or she may not be able to enroll in them ('not allowed'). Thus, the final recommendation value is a product of the raw value $r'_{i,j}$ and a function value $\mathit{reg}(i, j)$, where

$$\mathit{reg}(i,j) = \begin{cases} 1 & j \text{ is part of program} \\ 0 & j \text{ not allowed} \\ c_e(i) & \text{otherwise} \end{cases}$$

This introduces a further parameter $c_e(i) \in [0, 1]$ associated with courses that are not necessary to achieve the degree but may lead to an improvement of the final grade or may be interesting to the student. E.g. a student of bioinformatics may choose $c_e(i) = 0.5$ to get also recommendations for computer science courses that are not part of the bioinformatics

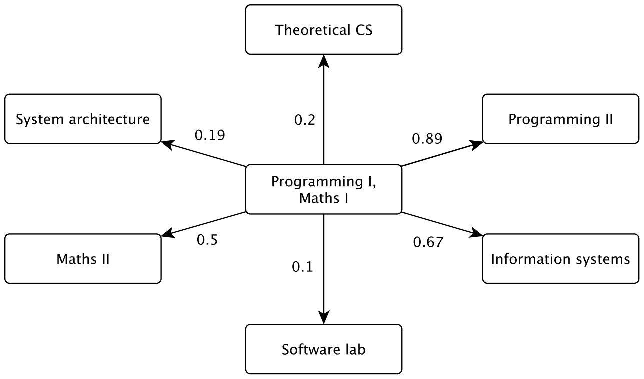


Figure 3: Example of a recommendation graph, based on the dependency graph given in 1. ‘Programming I’ and ‘Maths I’ have been passed already and the edge weights have been updated accordingly. The recommendation values were computed with $c_p = 0.76$, $c_g = 0.21$ and $c_m = 0.03$.

program. However, the default value is $c_e(i) = 0$.

Hence, the overall recommendation value of course j is

$$r_{i,j} = \big(c_p p_{i,j} + c_g(5 - \hat{g}_{i,j})/4 + c_m \deg^+(j)\big)\, reg(i,j) \quad (3)$$

with weight parameters by c_p, c_g, c_m.

To illustrate the influence of the different factors, we consider the following example. Suppose a first-year student in the winter semester uses the system to compose his first curriculum. We do not have any performance knowledge about the student, so this is a cold start scenario. Reconsider the dependency graph in Figure 1. Because of the high out-degrees, we recommend ‘Programming I’, ‘Maths I’ and ‘Theoretical CS’. The student successfully attends the first two of these courses in the following winter semester. Now we are able to incorporate the achieved grades in our prediction model. The now computed recommendation values per course are visualized as star graph shown in Figure 3. Finally a valid suggestion for the next semester based on the recommendation values is a combination of ‘Programming II’, ‘Information systems’ and ‘Maths II’. In general, at the beginning of every semester, we can provide the student with a personalized curriculum by compiling a list of lectures based on their recommendation score.

5.2 Evaluation

We now assess how similar our recommendation are to the actually selected courses of the students. Again, we separate the student data by semesters, such that recommendations are only based on data of previous semesters. To define the metric, let $\mathcal{T}$ be the set of semesters, $\mathcal{S}_t$ the set of students who took some course in semester $t \in \mathcal{T}$. Further, given some semester t, let $C^{i,t}_{sel}$ be the set of courses in which student i was enrolled and let $C^{i,t}_{rec}$ be the set of recommended courses for student i. We adopt a top-k recommendation policy in which we recommend only the k courses with the highest recommendation value. Moreover, we only take into account lectures which were available in the given semester and study program.

To approximate the conformity of our recommendations we consider the *conformity score*

$$1 - \frac{1}{|\mathcal{T}| + |\mathcal{S}_t|} \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{S}_t} \frac{\min(k, |C^{i,t}_{sel}|) - |(C^{i,t}_{rec} \cap C^{i,t}_{sel})|}{\min(k, |C^{i,t}_{sel}|)},$$

where the second term calculates the average ratio of the number of courses that have been selected by the student but were not recommended or that were recommended but not selected. So we end up with a score, indicating the congruency of our recommendations with the student’s actual course selections.

We evaluated the conformity score w.r.t. several combinations of the recommendation parameter values of Eq. (3). The considered recommendation sizes are 4 and 6 courses, since for most students this is a realistic balance between study progression and manageable a workload.Since we are interested in the relationship between the conformity score and the distribution of the parameters, in the first place we either fix c_p or c_g to 1 while the rest stays at zero which captures the performance of a single component of our approach. Moreover, we look for the best combination of both, course dependency graph (c_p) and grade prediction (c_g). The third parameter $c_m = 1 - c_p - c_g$ results from the choice of the first two, which makes the search two-dimensional.

Our results in Table 2 show that with increasing k the conformity grows as more courses are recommended. The first two columns of the table point out that the course dependency graph has a higher explanatory value for the recommendation than the grade prediction. A recommendation only based on the performance hardly achieves a value exceeding 50 percent while course dependency alone reaches 70 percent. Therefore it is clear that c_p has to be determined significantly larger than c_g. This observation is approved within the third column as in all top-k recommendations we reached the best conformity with $c_p \approx 0.76$, $c_g \approx 0.21$ and $c_m \approx 0.03$.

According to these scores our recommendations and the choices of the students have an average overlap of about 70 percent. Hence, there are recommended courses that the student did not choose. An example for this case is given by the core lecture ‘Embedded Systems’. We recommended this course to 89 students, while only 4 of them actually took the course in the corresponding semester. As opposed to mathematically demanding lectures such as ‘Complexity Theory’, which is only recommended for a small set of very strong students, this course seems to be a good choice for many students but is taken only by few. Moreover, in one semester the number of recommendations for basic courses was about 200 while only 90 students actually attended the courses. This could be related to the fact that many students withdraw from courses after a few weeks when they feel that the course is too demanding for them. In this case, the data does not show their trial for this course.

6. CONCLUSION

We proposed an approach that gives personalized course recommendations for students in order to improve the obtained grades and to decrease the time-to-degree. We combined a course dependency graph and performance predictions to

Table 2: The conformity score for different valuations of the recommendation value parameters (c_p, c_g, c_m) and different top-k recommendation policies.

top-k	$(c_p, c_g) = (1, 0)$	$(c_p, c_g) = (0, 1)$	$(c_p, c_g)^*$
4	0.5913	0.3857	0.6349
5	0.6580	0.4564	0.6962
6	0.7138	0.5326	0.7432

determine a recommendation value for each course. We assumed that only the top-k courses are given as a personalized curriculum for a student and tested their conformity to the actually selected courses of the student. This, however, does not indicate that our approach significantly improves the students' grades or time-to-degree as we expect that students do not make optimal choices.

An interesting insight from our results is that the course dependency graph seems better suited for course recommendation than grade prediction even though it is only based on aggregated information and does not consider any meta data. From this result it seems that students tend to focus more on a course ordering that older students established rather then selecting according to their own confidence or skill.Another interesting result is the large overlap (around 70 percent) of recommended and chosen courses. Moreover, some courses are not taken by students even though our model indicates that they would lead to an improvement in performance.

The model itself is flexible in the sense that one can easily adjust or extend it by changing the recommendation formula and/or incorporate more information to make the grade prediction more precise. A possible extension is the integration of more personalized information given by the student before calculating their recommendations. For example a student is more interested in practical lectures, so she uses an interface to let the system know. Thus, we would be able to give courses of this category a positive effect on their recommendation value. The challenge here is to separate the courses into appropriate categories, since the way a course is designed strongly depends on the lecturer and other factors.

To evaluate the system, it would be interesting to monitor a sufficiently large number of students during their studies that choose only recommended courses or at least is exposed to the course recommendations. An easier evaluation would be possible with a simulation of hypothetical student traces according to our grade prediction approach, where in each semester we assume that a student chooses only recommended courses.

7. REFERENCES

[1] R Asif, A Merceron, S Abbas Ali, and N Ghani Haider. Analyzing undergraduate students' performance using educational data mining. *Computers & Education*, 113:177 – 194, 2017.

[2] M Baron. *Probability and Statistics for Computer Scientists*. Chapman & Hall, 2014.

[3] CM Chen, CY Liu, and MH Chang. Personalized curriculum sequencing utilizing modified item response theory for web-based instruction. *Expert Systems with applications*, 30(2):378–396, 2006.

[4] J Duchi, E Hazan, and Y Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[5] C Hansen, C Hansen, N Hjuler, St Alstrup, and C Lioma. Sequence modelling for analysing student interaction with educational systems. In *Conference on Educational Data Mining*, pages 232–237, 2017.

[6] A Karatzoglou, X Amatriain, L Baltrunas, and N Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Conference on Recommender systems*, pages 79–86. ACM, 2010.

[7] A Polyzou and G Karypis. Grade prediction with models specific to students and courses. *International Journal of Data Science and Analytics*, 2(3-4):159–171, 2016.

[8] S Ray and A Sharma. A collaborative filtering based approach for recommending elective courses. In *International Conference on Information Intelligence, Systems, Technology & Management*, pages 330–339. Springer, 2011.

[9] Z Ren, X Ning, and H Rangwala. Grade prediction with temporal course-wise influence. *Conference on Educational Data Mining*, 2017.

[10] F Ricci, L Rokach, B Shapira, and PB Kantor. *Recommender systems handbook*. Springer, 2015.

[11] M Saarela and T Kärkkäinen. Analysing student performance using sparse data of core bachelor courses. *Journal of educational data mining*, 7(1), 2015.

[12] A Slim, GL Heileman, J Kozlick, and CT Abdallah. Employing markov networks on curriculum graphs to predict student performance. In *Machine Learning and Applications, Conference on*, pages 415–418. IEEE, 2014.

[13] SE Sorour, T Mine, K Goda, and S Hirokawa. A predictive model to evaluate student performance. *Journal of Information Processing*, 23(2):192–201, 2015.

[14] M Sweeney, J Lester, and H Rangwala. Next-term student grade prediction. In *Big data*, pages 970–975. IEEE, 2015.

[15] M Sweeney, H Rangwala, J Lester, and A Johri. Next-term student performance prediction: A recommender systems approach. *arXiv preprint arXiv:1604.01840*, 2016.

[16] A Töscher and M Jahrer. Collaborative filtering applied to educational data mining. *KDD cup*, 2010.

[17] H Vossensteyn, A Kottmann, B Jongbloed, F Kaiser, L Cremonini, B Stensaker, E Hovdhaugen, and S Wollscheid. Dropout and completion in higher education in europe: Main report. 2015.

[18] J Zimmermann, KH Brodersen, HR Heinimann, and JM Buhmann. A model-based approach to predicting graduate-level performance using indicators of undergraduate-level performance. *Journal of Educational Data Mining*, 7(3):151–176, 2015.

Who are they looking at? Automatic Eye Gaze Following for Classroom Observation Video Analysis

Arkar Min Aung
Worcester Polytechnic Institute
aaung@wpi.edu

Anand Ramakrishnan
Worcester Polytechnic Institute
aramakrishnan@wpi.edu

Jacob R. Whitehill
Worcester Polytechnic Institute
jrwhitehill@wpi.edu

ABSTRACT

We develop an end-to-end neural network-based computer vision system to automatically identify *where* each person within a 2-D image of a school classroom is looking ("gaze following"), as well as *who* she/he is looking at. Automatic gaze following could help facilitate data-mining of large datasets of *classroom observation* videos that are collected routinely in schools around the world in order to understand social interactions between teachers and students. Our network is based on the architecture by [27] but is extended to predict whether each person is looking at a target inside or outside the image; and to predict not only where, but who the person is looking at. Moreover, since our focus is on classroom observation videos, we collected a dataset from scratch of publicly available classroom sessions from 70 YouTube videos and collected labels from 408 labelers who annotated a total of $17,758$ gazes in $2,263$ unique image frames. Results of our experiments indicate that the proposed neural network can estimate the gaze target – either the spatial location or the face of a person – with substantially higher accuracy compared to several baselines.

Keywords

Automatic Eye Gaze Following; Classroom Observation Videos; Deep Neural Networks

1. INTRODUCTION

The nature and quality of teacher-student interactions in school classrooms are predictive of learners' development. Numerous observational studies and several causal studies have demonstrated the link between emotional and instructional support in the classroom and children's cognitive, social, and emotional skills [18, 23]. In order to discover how classroom interactions are related to learning outcomes, educational researchers often conduct *classroom observation* sessions, whereby human coders score either live or video-recorded classroom observations (typically 1 hour long each) along different dimensions, such as positive climate, teacher sensitivity, language modeling, quality of feedback, etc [25]. The Gates Foundation Measures of Effective Teaching (MET) project [16], in particular, recorded tens of thousands of hours of classroom observations across the United States with the aim of discovering best practices for how to teach students most effectively.

One of the major impediments to learning more from classroom observation video datasets is the difficulty and labor involved in coding them. Deep understanding of teacher-student interactions requires the coder to consider how the affective, linguistic, and pedagogical channels interact, and to interpret interactions within the context of classroom instruction. However, classroom observations contain multiple students and teachers interacting simultaneously in different parts of the classroom. It is easy for human coders to miss a subtle but important interaction. As a result, scores often can vary across coders, and multiple codes per video must be collected to obtain a reliable estimate. It would thus be invaluable to devise methods that could at least partially automate the process of classroom observation coding. Such a system could be useful not only for educational data-mining of large-scale classroom observation datasets, but also facilitate teachers' professional development by showing them video examples from their own classrooms in which they scored particularly high or low along different dimensions.

One important element of effective teacher-student interactions involves the students' and teachers' **eye gaze**: Does the teacher convey respect to his/her students by looking them in the eye when he/she is talking to them (*positive climate*)? Does the teacher notice when specific persons in the room are bored, confused, or even bullied (*teacher sensitivity*)? Tracking the eye gazes of students can also provide information on their thoughts and intentions [5] and may indirectly reveal how engaged they are in their learning.

In this paper, we take a tiny step towards creating an automated classroom observation scoring system. In particular, we build a prototype computer vision-based system for *automated eye gaze following* that estimates, for each person in the classroom, where she/he is looking. Such a system can be used to data-mine classroom observation video datasets. It could also facilitate "smart classrooms", which track gazes of both students and teachers, identify disengaged or distressed students, and help teachers to better recognize whether they are paying attention to the right thing or the right student in the classroom.

Deep learning for gaze following in classrooms: In this work, we explore a machine learning-based approach to automatic recognition of where a person in the image is looking. In particular, we build an end-to-end deep neural network that takes 2-D static images of multiple people as inputs and infers (x, y) coordinates of where *each* person

This material is based upon work supported by the National Science Foundation under Grant No. #1551594 and Spencer Small Research Grand No. #201800131.

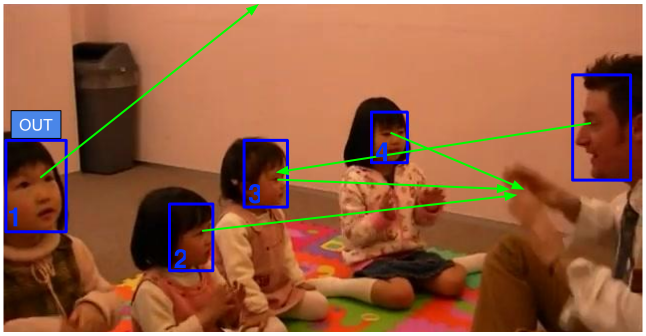


Figure 1: Eye gaze targets labeled by a human labeler for each person in the image. Labelers also indicate targets that are located outside the field-of-view (indicated by "OUT"). Can we build a computer vision system that can estimate *where* each person is looking? In this image, the man is looking at child #3. Can we identify automatically *who* each person is looking at? Image from `https://goo.gl/xUdYbC`

is looking at in the image as outputs. This computational problem is known as *gaze following* [10]. Gaze following from 2-D images is particularly challenging since 1) no additional information of the scene, such as depth information, is available and a person can be looking at any of the different planes of depth in the image, 2) people in the image can be looking at objects either inside the image or outside the image, 3) the eyes of some people may be blurred or partially invisible. Nonetheless, requiring only 2-D images is attractive because of the ubiquity and greater convenience of using commodity 2-D cameras. Our automated system is based on the architecture by [27], who tackled a similar problem for general images from the web. However, our approach differs from theirs in several ways, including the prediction outputs, deep neural network architectures, training techniques, dataset collection, and application focus.

Contributions: (1) We explore a deep learning-based architecture, based on related work by [27], for automatic eye-gaze following from 2-D images of classroom observation videos. (2) We extend the model of [27] to support gaze targets that can be *outside* the camera's field-of-view. Especially due to the lack of depth information, this is a highly challenging problem, both for human labelers and the machine. (3) Our application focus is on school classrooms, which contain many subjects (not just a few, as in [27]), who gaze not only inside but sometimes also outside the field-of-view. We thus collected and annotated (see Figure 1) a new dataset of images from classroom videos. (4) Since classroom observation analysis is largely about interaction between subjects, we explore the accuracy of our automatic gaze following system in identifying which *face* (not just object) each person is looking at. Detailed methodology and results for contribution (1), (2) and (3) are described in Section 3 and those of contribution (4) are described in Section 5.

2. RELATED WORK

Eye gaze following: Due to the importance of following gaze of others, which humans do naturally when communicating, collaborating and socializing, researchers in the field of robotics, computer vision and machine learning have recently started to formulate and tackle the problem of automatic gaze following within different contexts: In some settings [15, 3, 11], there is only a single person whose gaze is being followed, e.g., a student who is interacting with a mobile phone or a tablet [19] to play an educational game [31]. In other settings (such as ours), the camera examines an entire scene containing many people, and the gaze of *each* person in the scene is followed [24] [21] [28] [27]. While most of the prior work uses RGB data, some approaches also use depth information [24]. More recently, researchers have considered gaze following not only from static images but also how to harness temporal information from an entire video to better estimate the person's gaze target [28]. In this work, we only consider gaze following from static 2-D images extracted from classroom observation videos but future work can explore following gaze by using temporal information from a sequence of images.

Saliency modeling: Gaze following is related to saliency modeling, whereby image features of different levels of abstraction (low-, mid-, and high-level) are examined to consider the most likely locations in the image to which an observer would visually attend [15]. [3] made a connection between these two by stating that an observer looking at an image containing people may follow the gaze of people rather than actually fixating on salient objects in that image. Therefore, gaze following can play a complementary role in solving the problem of saliency model of attention. [7] explored the problem of predicting a driver's gaze behaviours and identifying the attention of a driver by detecting saliency in a complex driving environments.

Modeling non-verbal cues of students and teachers: There has been substantial prior work on analyzing learners' affective states from video using computer vision [17, 12, 4, 30]. Much of this work has focused on intelligent tutoring systems. More recently, researchers in multi-modal machine learning and educational data mining have investigated how to characterize the dynamics of an entire classroom. For example, [9, 8] explored approaches for segmenting and recognizing students' and teachers' speech in unconstrained classrooms based on different configurations of Microsoft Kinect cameras. For automated classroom observation scoring (e.g., of CLASS [25]), we are only aware of one prior work: [26] developed a computer vision system, optimized within a multiple-instance learning framework [22], to estimate which 3-minute snippets of classroom videos were most relevant for CLASS coders to watch.

3. EXPERIMENT I: METHODOLOGY

3.1 Data collection

Since the application focus of our study is gaze following in *school classrooms*, we collected our own dataset of classroom observation sessions. In particular, we harvested 70 videos publicly available on YouTube of school classrooms. The study was approved under WPI IRB 18-0101. In contrast to publicly available annotated data on gaze following (the only such dataset of which we are aware is GazeFollow [27]), classroom observation videos often contain *many* people per image frame, and the kinds of background clutter differ significantly from that of GazeFollow, which largely consists of images used for more general object detection research.

From each video in our collection, we extracted 1 frame approximately every 10 seconds. After extracting frames from videos, we used Faster R-CNN for face detection [14] to obtain face bounding boxes (top left (x, y) coordinate, width and height) in extracted frames.

Annotation: Ground-truth gaze annotations from the image frames were collected using at least 3 labelers per image on Amazon Mechanical Turk (AMT). Labelers used an online annotation tool that we custom-built for this work, using JavaScript and HTML5, to annotate two main components of each subject in each scene. The first component is to identify the gaze target for each person (identified automatically by the face detector as described above) which is indicated by a line, starting between the eyes of a person and ending on an object or a person which the person is attending to. The second component is the indication of whether the person is looking at something inside or outside the image. We collected three gaze annotations each for $17,758$ faces in $2,263$ images, resulting a total of $48,907$ gaze annotations from 408 unique annotators.

3.2 Approach

Using the datasets annotated on AMT, our goal is to build a convolutional neural network (CNN) which takes in the whole image of the scene and predicts the gaze target of each person in the image along with the indication of whether that target is inside or outside the image. We have observed from our annotated datasets that predicting gaze can be ambiguous. If there are multiple people or several salient objects in the image, or the eyes of individuals in the image are not clearly visible, human labelers may disagree when predicting gaze locations. Due to this inherent uncertainty in the problem, we explore various options to design our model to support multimodal predictions.

We can formulate gaze following as either a regression or a classification task. **Regression**: the network regresses to (x, y) coordinates of the gaze target of each person in the image using the Euclidean distance between the predicted and ground-truth as the cost function. The disadvantage of using regression is that our predictions are constrained to be unimodal. Since each face in each image was labeled by multiple annotators, we can define the ground-truth by either (a) computing the mean (x, y) location over all labels per face, or (b) treating each location as a separate label. **Classification**: the gaze location is quantized into one cell on an $N \times N$ grid, and the network's job is to choose the correct cell for each person in the image. As the cost function, we can use cross-entropy loss. Classification naturally supports multimodal outputs since multiple gaze annotations at different cells can be treated as soft labels [1]. The disadvantage of this approach is that the choice of grid size can affect the precision of predictions (i.e. smaller numbers of grid cells N will result in poor precision). Another issue is that cross-entropy loss does not gradually penalize mistakes based on distance – misclassification which is off by one grid cell is penalized just as much as misclassification which is off by several cells on a grid.

3.3 Architecture

The deep learning architecture is based on the model by [27] and is depicted in Figure 2. The gaze target for each person in the image is predicted independently based on two information sources: close-up information of the person's face (automatically detected by a separate face detection network [14]), and the whole image. Each information source is processed by a separate pathway consisting of a CNN, and the pathways' predictions about the person's gaze target are merged at the end. We call the combined architecture the *Merged Model*. In contrast to [27], we use the VGG16 [29] as the backbone of each CNN since we found empirically that it performed better than AlexNet [20]. Two other differences from [27] are the network optimization techniques and the use of multi-task learning (as described in Section 3.4). **Inputs**: The inputs of the Merged Model are a cropped, close-up face image (64×64 pixels); the $(r, c) \in N \times N$ location of the center of the person's head in the image; and the resized 256×256 pixels image of the whole frame. We chose $N = 8$ in our experiments. **Outputs**: For regression, the gaze target is represented as an (x, y) coordinate pair. For classification, the gaze target consists of a 1-hot vector indicating which of the $N \times N$ grid cells contains the gaze target. In addition (for both regression and classification), the network also contains an "in"/"out" binary prediction of whether the gaze target is inside or outside the image.

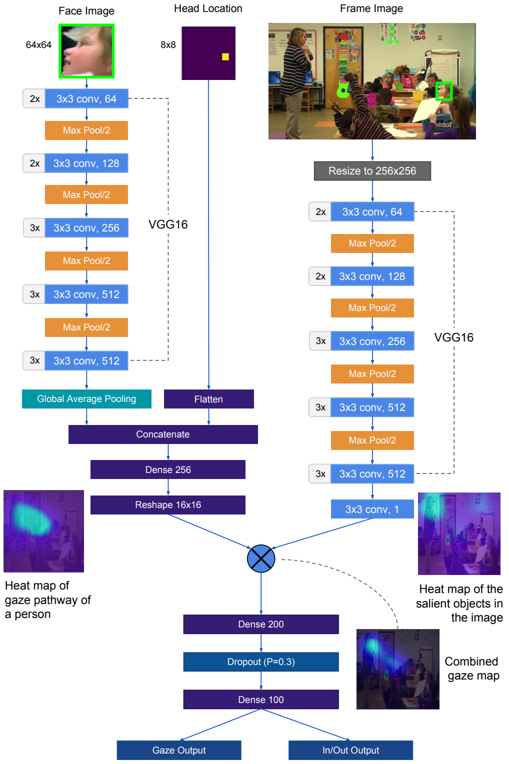


Figure 2: Deep neural network architecture, based on [27], for automatic eye-gaze following in school classrooms, consisting of two independent prediction pathways.

The intuition behind the Merged Model is that two CNNs are trained to solve two subproblems in a fully end-to-end fashion with only the gaze location and the "in"/"out" label as supervision to the model: (1) The close-up face CNN (left pathway in Figure 2) implicitly estimates the head pose and the direction of the gaze of the subject in order to produce a heat map (shown as `Reshape 16×16` in Figure 2) of where the person is looking. In the figure, the heat map roughly shows a "cone" of possible gaze targets to the upper-left of the child's head. (2) The frame-image CNN (right pathway in Figure 2) identifies the salient objects in the image. This network has access to the entire original image but does not know the location of the subject. In the figure, the salient object heat map highlights the teacher in the upper-left of the image. In [32], the authors showed that objects tend to emerge in the filter kernels of the deep layers of CNNs; therefore, we take a filter kernel at the end of the right pathway (shown as 3×3 `conv, 1` in Figure 2). This produces the heat map of salient objects in the original image. Each heat map from each branch is combined by element-wise multiplication.

3.4 Training procedure

Data partitions: The 70 YouTube videos containing school classrooms were partitioned into training (12,430 gazes), validation (2,664 gazes), and testing (2,664 gazes) sets, such that none of the frames from any video was assigned to more than one set. The validation set was used for early stopping. The accuracy on the test set can be considered a performance estimate on faces that the network has never seen before.

Optimization: We used the following procedure for both the regression and classification formulations: We first performed transfer learning by initializing both CNNs with weights pre-trained on ImageNet [29]. We augmented the classroom images from our dataset by flipping the original images (frame image pathway) as well as the individually cropped face images, head locations and gaze locations (face pathway) left to right. We trained the final Merged Model first by freezing all the convolutional layers and training only the fully connected layers with RMSProp [13] (learning rate = 0.01, $\rho = 0.9$). Then all the previously frozen convolutional layers were unfrozen and the model was fine-tuned with SGD with momentum (learning rate=0.0001, momentum=0.9). The model was trained until there was no improvement in validation loss.

Multi-task learning: Since the Merged Model predicts the location of the gaze in the image as well as "in"/"out", it is performing multiple tasks, and we can use multi-task learning (MTL) [6] for training. Sharing the same hidden layers to solve several tasks forces the model to find representations which capture all of the tasks and thus reduce the risk of overfitting [2]. We found empirically that MTL helped to reduce overfitting and improve prediction accuracy. Table 1 compares the performance of the Merged Model with and without MTL. With MTL, the cross-entropy loss for both the grid output and the In/Out output is higher (worse) on the training set, but lower (better) on the testing set, compared to training two networks to handle each task separately. We thus adopted the MTL approach for training.

3.5 Accuracy measurement

Accuracy is measured for predicting the gaze target of each person (identified automatically by a face detector [14]) in each extracted frame from each of the YouTube videos (see Section 3.1). For **classification** of the gaze target among the $N \times N$ grid cells, we evaluated accuracy in terms of the cross-entropy (CE) loss w.r.t. the label distribution induced by the 3 annotators per example. For **regression** to an (x, y) location, we use mean absolute error (MAE), mean Euclidean distance and mean angular error (between the center of the person looking to their gaze target) in degrees, where the ground-truth is defined as the *average* annotation over all the annotators. In addition (for both regression and classification), we also used the Area Under the Receiver Operating Characteristics Curve (AUC) to evaluate the binary classification of whether the target is inside or outside the field-of-view.

Table 1: Effects of multi-task learning. CE Loss refers to Cross Entropy Loss and reported values are Cross Entropy Loss of Merged Model predicting gaze on 8×8 grid.

	Only grid output	Only In/Out output		Both grid output and In/Out output		
	CE Loss	CE Loss	AUC	CE Loss (Grid Output)	CE Loss (In/Out)	AUC (In/Out)
Training	3.27	0.32	0.63	3.39	0.33	0.60
Testing	3.59	0.46	0.59	3.58	0.43	0.62

3.6 Baseline comparison

When assessing the accuracy of any neural network, it is important to establish the relevant baselines for comparison. For classification, we use a uniform distribution over all $N \times N$ grid cells – in other words, a random guess in the whole image as to where the person is gazing. Alternatively, we can assume a center prior (motivated by [15]), consisting of the center 2×2 grid cells over the $N \times N$ grid. A variation on the center prior is to place a 2-D Gaussian – whose standard deviation σ is optimized directly on the *test set* for best possible accuracy – centered on the middle of the image, and assign probabilities to the $N \times N$ cells based on the Gaussian probability density function. For regression, we use a center prior corresponding to the midpoint in the image; we also compare to randomly selected points in the image.

As stronger baselines, we also consider linear regression to analyze the vectorized face pixels concatenated with head locations to predict (x, y) coordinates, as well as logistic regression to predict cells on a $N \times N$ grid. Finally, as a way of understanding which part of the Merged Model contains more information, we also compare to a Face-to-Gaze model consisting of a CNN that takes a cropped, close-up face image and location of head in the image as inputs, and predicts the location of the gaze in the image as well as "in"/"out" – this is the left pathway of Figure 2. Comparing with this baseline helps us understand how much the saliency pathway improves performance.

4. RESULTS I

Accuracy results on test images of the Merged Model compared to the baselines are shown in Table 2 (for regression) and Table 3 (for classification). Our Merged Model achieves mean Euclidean distance of 69.82 pixels on 256×256 pixel image (for regression) and cross entropy loss of 3.5855 on 8×8 grid (for classification) for gaze locations. These numbers are better than for the random gaze, center prior, center Gaussian, linear and logistic regression baselines. For comparison, human labelers exhibited a mean Euclidean distance of only 41.04 pixels on 256×256 pixel image, which

Table 2: Regression accuracy of the Merged Model for predicting the (x, y) location (within a 256×256 image) of where each person in each classroom image is looking. Accuracy is compared to human annotators and three baseline models.

	MAE	Mean Euclidean Distance	Mean Absolute Angular Error	AUC for In/Out
Random Gaze	79.74	124.15	67.24°	-
Center Region	52.76	82.11	48.36°	-
Linear Regression	49.63	77.34	55.21°	-
Face-to-Gaze	45.74	71.53	39.91°	0.54
Merged Model	**44.49**	**69.82**	**38.30°**	**0.62**
Human	25.91	41.04	18.38°	0.70

Table 3: Classification results on 8×8 grid of the Merged Model compared to several baselines.

	Cross Entropy Loss (Grid Output)	AUC for In/Out
Center Gaze (Center 4 cells)	15.8047	-
Uniform Gaze	4.1589	-
Center Gaussian	4.0561	-
Logistic Regression	3.9997	-
Face-to-Gaze	3.7511	0.5459
Merged Model	**3.5855**	**0.6223**

is a bit more than half the error of the Merged Model, indicating that the machine's accuracy still has much room for improvement.

For classifying whether the gazes end inside or outside the image, the Merged Model achieved an AUC of 0.62, whereas humans scored 0.70 on the same task. The relatively low human accuracy suggests that detecting whether a person is looking inside or outside the image is quite challenging in the classroom images.

Figure 3 shows qualitative results of some of the gaze predictions (represented by thick yellow arrows) by Merged Model. It can be seen that the model makes decent predictions on the general direction of gazes but sometimes misses the endpoints on salient objects in the scene. In Figure 3, three girls in the middle are looking at the man's hands but the gaze predictions end before the hand.

One notable fact is that the **Face-to-Gaze** model's performance is very similar to the Merged Model's performance. This suggests that our Merged Model is predicting gaze locations mainly by using the head pose and gaze pathway of the subject and less on the salient objects in the image. One possible explanation is that our dataset does not contain enough variety of classroom environments for the model to learn how to identify salient objects in classroom images.

5. EXPERIMENT II: WHO ARE THEY LOOKING AT?

We use the same neural network depicted in Figure 2 to predict *who* each person is looking at. This is especially useful in school classrooms, in which both students and teachers are often looking at other *people*, not just objects. Specifically, we use the *classification* approach to predict which of the $N \times N$ grid cells each person is gazing at. The face contained within that cell is then predicted to be target face of that person's gaze. We note that, depending on the grid size and the specific image, multiple faces might appear within the same cell. A principled approach to handle to this issue would be to distribute the probability mass output by the neural network among all the faces within that cell in proportion to the size of each face. However, in this exploratory study, we simply assume that no grid cell contains more than 1 face.

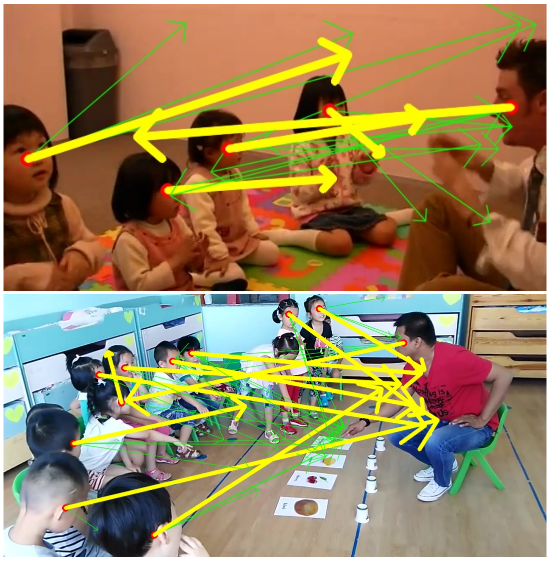


Figure 3: Qualitative results of gaze predictions by our Merged Model on the test set. Thin green arrows are ground truth annotations. Since there are multiple gaze annotations for each individual, there are multiple green arrows for each individual. Thick yellow arrows are predictions by Merged Model. Images (top to bottom) taken from: `https://goo.gl/xUdYbC`, `https://goo.gl/pcwQ5P`

5.1 Methodology

First, we computed the subset of all people in all image frames of our original YouTube dataset in which all annotators agreed that the person is looking at another *face* (not just another object somewhere in the image). Note that the labelers can still differ as to which particular face the person is looking at. By doing so, we obtained, 410 faces where all labelers agree that the person is looking at another face out of 17, 759 faces in our dataset. On the same data subset, we use the Merged Model to compute the softmax probabilities across all $N \times N$ grid cells of where each person was looking. From these probability outputs (for each person in each image), we remove every cell that does not contain any face (as determined by the face detector) and renormalize. We then choose the grid cell with the highest probability as the face that the person is most likely to be gazing at.

In order to evaluate how well our network is performing on determining which face a person is looking at, we took the top 1 face, top 2 faces, and top 3 faces. For the top-1 face, we choose the grid cell with the highest probability as the face that the person is most likely to be gazing at as predicted by the deep neural network. For top-2 and top-3 faces, if any of the top-2 and top-3 faces predicted by the network is the actual face which is agreed by the majority of human labelers, the prediction is regarded as a correct prediction.

As baselines, we can consider that the average number of faces (detected by the face detector [14]) per image was 6.87 on test set; hence, the baseline guess rate is $1/6.87 \approx 0.15$ for the test set. Moreover, we can estimate human accuracy in a leave-one-labeler-out fashion: for each unique labeler, in the subset of the dataset where all labelers agree that a person being annotated is looking at another face, we compare the face that the current labeler chooses with the face which the majority of other labelers agree on. In this fashion, we compute the accuracy (% correct) of the l^{th} labeler w.r.t. the other $l-1$ labelers. We then average across all labelers in our dataset. By doing so, we achieve the human level performance on determining whom the person is looking at in the classroom given that the person is looking at *a face*.

In order to make equal comparison with Merged Model's predictions, which is done on 8×8 grid, human annotations are quantized to cells on 8×8 grid and probability of one labeler agreeing with the rest of the labelers that a person being annotated is looking at a *specific* face (last row of Table 4).

6. RESULTS II

The results on test images, shown in Table 4, indicate that the Merged Model can predict the face target of people's eye gazes with substantially higher accuracy than just randomly guessing among all grid cells (8×8 grid) in the image containing faces. To put these results in context: if each classroom image contains 6.87 faces on average (as reported above), then the probability of 0.79 for $k = 3$ suggests that an automated gaze following system can usually determine at least which *group* of students a teacher is looking at. Interestingly, the accuracy of the Merged Model is close to that of human labelers when top 3 predicted faces are considered but still have room for improvement when only top 1 face is chosen.

7. CONCLUSION AND FUTURE WORK

The results in this paper indicate that an automatic neural network, based on the approach by [27] that analyzes 2-D images of school classrooms can estimate the gaze target location of each person in the image with accuracy substantially higher than chance and better than several other baselines as well. Moreover, the same architecture can be used to identify *who* each person is looking at more accurately than random guessing.

Future work: The most critical next steps are to (1) improve accuracy by collecting more training data and improving the accuracy of the annotations. (2) Given an improved eye gaze following system, we can begin to explore how automatic gaze estimates can be used to predict specific aspects of classroom observation protocols; for instance, the *positive climate* dimension of the CLASS is based explicitly (in part) on whether the teacher looks at his/her students [25]. Finally, (3) since multiple people often look at the same person (e.g., the teacher) in school classrooms, we will also investigate whether accuracy can be improved by estimating the gaze targets of all classroom participants *jointly* rather than separately.

Table 4: Probability of the Merged Model correctly identifying which face a person is looking at on 8×8 grid.

Top k faces	$k = 1$	$k = 2$	$k = 3$
Random Face	0.15	0.30	0.45
Merged Model	**0.47**	**0.65**	**0.79**
Human	0.82		

8. REFERENCES

[1] Aung, A. M., and Whitehill, J. R. Harnessing label uncertainty to improve modeling: An application to student engagement recognition. In *IEEE Automatic Face & Gesture Recognition* (2018).

[2] Baxter, J. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine learning 28*, 1 (1997).

[3] Borji, A., Parks, D., and Itti, L. Complementary effects of gaze direction and early saliency in guiding fixations during free viewing. *Journal of vision 14*, 13 (2014).

[4] Bosch, N., D'Mello, S., Baker, R., Ocumpaugh, J., Shute, V., Ventura, M., Wang, L., and Zhao, W. Automatic detection of learning-centered affective states in the wild. In *International conference on intelligent user interfaces* (2015).

[5] Brooks, R., and Meltzoff, A. N. Gaze following: A mechanism for building social connections between infants and adults. *In Mechanisms of social connection: from brain to group* (2014).

[6] Caruana, R. Multitask learning: A knowledge-based source of inductive bias. In *International Conference on Machine Learning* (1993).

[7] Deng, T., Yang, K., Li, Y., and Yan, H. Where does the driver look? top-down-based saliency detection in a traffic driving environment. *IEEE Transactions on Intelligent Transportation Systems 17*, 7 (2016).

[8] D'Mello, S. K., Olney, A. M., Blanchard, N., Samei, B., Sun, X., Ward, B., and Kelly, S. Multimodal capture of teacher-student interactions for automated dialogic analysis in live classrooms. In *ACM international conference on multimodal interaction* (2015).

[9] Donnelly, P. J., Blanchard, N., Samei, B., Olney, A. M., Sun, X., Ward, B., Kelly, S., Nystrand, M., and D'Mello, S. K. Multi-sensor modeling of teacher instructional segments in live classrooms. In *ACM international conference on multimodal interaction* (2016).

[10] Emery, N. J. The eyes have it: the neuroethology, function and evolution of social gaze. *Neuroscience & Biobehavioral Reviews 24*, 6 (2000).

[11] Fathi, A., Hodgins, J. K., and Rehg, J. M. Social interactions: A first-person perspective. In *Computer Vision and Pattern Recognition* (2012).

[12] Grafsgaard, J., Wiggins, J. B., Boyer, K. E., Wiebe, E. N., and Lester, J. Automatically recognizing facial expression: Predicting engagement and frustration. In *Educational Data Mining* (2013).

[13] Hinton, G. Rmsprop: Divide the gradient by a running average of its recent magnitude.

[14] Jiang, H., and Learned-Miller, E. Face detection with the faster r-cnn. In *IEEE Automatic Face & Gesture Recognition* (2017).

[15] Judd, T., Ehinger, K., Durand, F., and Torralba, A. Learning to predict where humans look. In *International Conference on Computer Vision* (2009).

[16] Kane, T. J., McCaffrey, D. F., Miller, T., and Staiger, D. O. Have we identified effective teachers? validating measures of effective teaching using random assignment. In *Research Paper. MET Project. Bill & Melinda Gates Foundation* (2013).

[17] Kapoor, A., Burleson, W., and Picard, R. W. Automatic prediction of frustration. *International journal of human-computer studies 65*, 8 (2007).

[18] Kontos, S., and Wilcox-Herzog, A. Teachers' interactions with children: Why are they so important? research in review. *Young Children 52*, 2 (1997).

[19] Krafka, K., Khosla, A., Kellnhofer, P., Kannan, H., Bhandarkar, S., Matusik, W., and Torralba, A. Eye tracking for everyone. In *Computer Vision and Pattern Recognition* (2016).

[20] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (2012).

[21] Marín-Jiménez, M. J., Zisserman, A., Eichner, M., and Ferrari, V. Detecting people looking at each other in videos. *International Journal of Computer Vision 106*, 3 (2014).

[22] Maron, O., and Lozano-Pérez, T. A framework for multiple-instance learning. In *Advances in neural information processing systems* (1998).

[23] Mashburn, A. J., Pianta, R. C., Hamre, B. K., Downer, J. T., Barbarin, O. A., Bryant, D., Burchinal, M., Early, D. M., and Howes, C. Measures of classroom quality in prekindergarten and children's development of academic, language, and social skills. *Child development 79*, 3 (2008).

[24] Mukherjee, S. S., and Robertson, N. M. Deep head pose: Gaze-direction estimation in multimodal video. *IEEE Transactions on Multimedia 17*, 11 (2015).

[25] Pianta, R. C., La Paro, K. M., and Hamre, B. K. *Classroom Assessment Scoring System^{TM}: Manual K-3.* Paul H Brookes Publishing, 2008.

[26] Qiao, Q., and Beling, P. A. Classroom video assessment and retrieval via multiple instance learning. In *International Conference on Artificial Intelligence in Education* (2011).

[27] Recasens, A., Khosla, A., Vondrick, C., and Torralba, A. Where are they looking? In *Advances in Neural Information Processing Systems* (2015).

[28] Recasens, A., Vondrick, C., Khosla, A., and Torralba, A. Following gaze in video. In *Computer Vision and Pattern Recognition* (2017).

[29] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[30] Wang, Z., Pan, X., Miller, K. F., and Cortina, K. S. Automatic classification of activities in classroom discourse. *Computers & Education 78* (2014).

[31] Zain, N. H. M., Razak, F. H. A., Jaafar, A., and Zulkipli, M. F. Eye tracking in educational games environment: evaluating user interface design through eye tracking patterns. In *International Visual Informatics Conference* (2011).

[32] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856* (2014).

Accurate Measurement of Lexical Sophistication with Reference to ESL Learner Data

Ben Naismith
University of Pittsburgh
Department of Linguistics
4200 Fifth Ave, Pittsburgh, PA 15260
1-412-624-5900
bnaismith@pitt.edu

Na-Rae Han
University of Pittsburgh
Department of Linguistics
4200 Fifth Ave, Pittsburgh, PA 15260
1-412-624-5900
naraehan@pitt.edu

Alan Juffs
University of Pittsburgh
Department of Linguistics
4200 Fifth Ave, Pittsburgh, PA 15260
1-412-624-5900
juffs@pitt.edu

Brianna Hill
University of Pittsburgh
School of Computing and Information
4200 Fifth Ave, Pittsburgh, PA 15260
1-412-624-5900
blh82@pitt.edu

Daniel Zheng
University of Pittsburgh
Department of
Electrical and Computer Engineering
4200 Fifth Ave, Pittsburgh, PA 15260
1-412-624-5900
daniel.zheng@pitt.edu

ABSTRACT

One commonly used measure of lexical sophistication is the Advanced Guiraud (AG; [9]), whose formula requires frequency band counts (e.g., COCA; [13]). However, the accuracy of this measure is affected by the particular 2000-word frequency list selected as the basis for its calculations [27]. For example, possible issues arise when frequency lists that are based solely on native speaker corpora are used as a target for second language (L2) learners (e.g., [8]) because the exposure frequencies for L2 learners may vary from that of native speakers. Such L2 variation from comparable native speakers may be due to first language (L1) culture, home country teaching materials, or the text types which L2 learners commonly encounter. This paper addresses the aforementioned problem through an English as a Second Language (ESL) frequency list validation. Our validation is established on two sources: (1) the New General Service List (NGSL; [4]) which is based on the Cambridge English Corpus (CEC) and (2) written data from the 4.2 million-word Pitt English Language Institute Corpus (PELIC). Using open-source data science tools and natural language processing technologies, the paper demonstrates that more distinct measurable lexical sophistication differences across levels are discernible when learner-oriented frequency lists (as compared to general corpora frequency lists) are used as part of a lexical measure such as AG. The results from this research will be useful in teaching contexts where lexical proficiency is measured or assessed, and for materials and test developers who rely on such lists as being representative of known vocabulary at different levels of proficiency. This research applies data-driven exploration of learner corpora to vocabulary acquisition and pedagogy, thus closing a loop between educational data mining and classroom applications.

Keywords

Advanced Guiraud, corpus linguistics, English as a Second Language, ESL, learner corpora, lexical sophistication, vocabulary lists

1. INTRODUCTION

An enduring concern of researchers in second language (L2) vocabulary development is the basic set of words learners should know; moreover, having acquired this vocabulary, what kinds of intervention are best for promoting acquisition of the additional words that learners need in order to function professionally and academically [8, 23]? Thus, establishing the correct set of basic words that learners already know is important to be able to measure subsequent development in productive vocabulary knowledge. In order to accurately track the acquisition of new vocabulary over time, researchers have focused on quantitative measures that can be used to examine different aspects of the 'lexical richness' of learner output, including *lexical diversity*, which uses text internal measures such as VocD (D) and MTLD (e.g., [17, 21]); *lexical sophistication*, which makes reference to frequencies in corpora with measures like the Advanced Guiraud (AG) (e.g., [10, 28]); and *lexical depth*, which measures knowledge of usage (e.g., [6, 11]). In this paper, we focus on lexical sophistication because (1) the calculation of AG depends on the establishment of the correct set of high-frequency words that the learners may (already) know; (2) the frequency bands of 3000-9000 words are lexical items that researchers advocate should be the focus of instruction [25]; and (3) teacher perceptions of lexical proficiency have been shown to correlate strongly with lexical sophistication [10].

2. LITERATURE REVIEW

Vocabulary knowledge in a second language is a vital component in the development of L2 proficiency [23]. As a result, accurate measurement of vocabulary is important for all language learning stakeholders including learners, teachers, material developers, developers of standardized tests, and educational institutions. One common context of English as a Second Language (ESL) learning, and that of this study, is in tertiary education intensive English programs (IEPs). Most students entering IEPs already know some English, typically placing at the low-intermediate level and above. As a corollary, learners are expected to already know high-frequency English vocabulary such as the first 2000 words of the New General Service List (NGSL; [4]).

The stakes are high in that most students have a short time to prepare for academic work, and as such, the targeting of instruction to students' needs is important. Yet, this task is difficult for teachers because the first languages (L1s) of the students vary, and students may in fact not know all of the basic words assumed by frequency lists of basic vocabulary. Such lack of certainty makes measuring vocabulary development beyond the basic list challenging because at the higher levels learners may not be given credit for acquiring high-frequency words they are assumed to know, but in fact do not control in their productive lexicon. In contrast, low-frequency words that they already know, based on their own cultural or educational background, may wrongly be treated as newly acquired. This issue reflects a general concern that materials written for learners may not consider broader linguistic needs of the students [18] and that frequencies from large corpus analyses may not always reflect linguistic challenges (e.g., [16]).

The literature on vocabulary development has shown that Advanced Guiraud (AG) can be an effective method of measuring of lexical sophistication [12, 19], but may not always reflect development [11]. In essence, AG is a form of Type/Token ratio (TTR) [28] with two key differences. First, it takes as the denominator the square root of the total tokens, a measure designed to neutralize TTR's sensitivity to text length. Second, types that are very frequent, for example the 2000 most frequent words on the NGSL, are removed from the total types [28, 12]. As a result, AG incorporates frequency information, while other measures do not.

In [12], Daller and Xue compared two groups of Chinese-speaking learners, one in China and the other in the UK. They found that Guiraud (all types/√tokens) and AG were both effective at distinguishing the China group from the UK group, whose mean (stdev) AG scores were 0.72 (.2) and 0.94 (.29) respectively. However, when Daller et al. [11] investigated the longitudinal development of 42 Arabic-speaking ESL learners, the values of AG were low and increased minimally, ranging from an average of about 0.20 to 0.25 [11]. In neither study was the composition of the AG list of 2000 basic types specified, referred to only as 'the 2000 frequency band.' Considering, as [16] says, that the needs of the users should be accounted for when replicating a word list, knowing such information would be of great use to researchers seeking to evaluate and replicate previous results.

Supporting Daller and Xue's findings, Juffs [19] analyzed a subset of the Pitt English Language Institute Corpus (PELIC) data. He found that AG (using the 2000 frequency bands of the BNC-COCA at http://lextutor.ca as a lexical sophistication metric) was a better measure than D (a lexical diversity metric) in distinguishing progress in lexical development of Arabic, Chinese, and Korean learners who studied throughout the upper-intermediate (level 4) and advanced (level 5) levels in the Pitt IEP. Juffs found that the level 4 learners' AG scores ranged from 1.32 to 1.53 on average, whereas the level 5 learners' scores ranged from 1.90 to 2.12. However, Juffs' study, while suggestive, only included 254,055 tokens and did not fully utilize PELIC's written sub-corpus which actually consists of more than 4.2 million tokens when all L1s are included.

The studies reviewed here demonstrate large variability in terms of how frequency data are measured and collected. Not only are the 2000-word lists for AG inconsistent or unknown across studies, but so too is the definition of the 'types' which form the basis of many lexical measures. Although a full discussion of this area is beyond the scope of this paper (see, e.g., [22]), it directly impacts all measures using frequency lists. On one end of the spectrum, measurements such as TTR count types mechanically without grouping different forms in anyway, so that 'dog' and 'dogs' would be counted as two distinct types. In this approach, the value lies in the ease with which data can be analyzed automatically with no need for human judgements. However, should a learner who produces 'mango' and 'mangos' be said to have the same lexical range as someone who produces 'mango' and 'pomegranate', or can we assume that the latter student will also know the plural forms?

At the other extreme, many researchers (e.g., [1]) advocate for *word families* to be the base counting unit, i.e., a word plus its derivational and inflectional forms. For example, 'happy', 'happiness', 'unhappy', and comparative 'happier', would be one unit. While this solves the previous issue, it means that a learner would not be given credit for knowing words related by derivation to a common word, with, for example, 'actresses', 'actionable', and 'inaction' all belonging to the word family 'act' (http://lextutor.ca).

A third 'middle ground' approach advocated by Schmitt [24] uses lemmas as a measurement unit. A lemma typically refers to a word plus its inflected forms only; lemma information has accompanied various resources, including the Brown Corpus and the New-GSL (not to be confused with the NGSL) [3]. Thus, 'act', 'acted', and 'acting' would be one unit, but 'act' and 'actionable' separate units. In sum, when creating a word list there are numerous decisions to make regarding not only the relative value of word frequency, range, and dispersion, but even the unit of counting must be considered and justified [16].

Given the challenges in data collection and analysis, the lack of consensus as to best practice is unsurprising. Comparisons across studies are further complicated by small sample sizes, limited L1 backgrounds, and different learning contexts, all of which threaten the external validity and thus the generalizability of the results. The reported scores in this literature do, however, give this study a range of reasonable AG scores that one might expect.

In contrast, PELIC is a multi-million-word learner corpus representing learners from different L1 backgrounds who have studied together in the same location, using similar materials, and in the same educational context. Exploiting this unique dataset, we seek to address the following research questions:

(1) How can data mining tools be applied to a learner corpus to produce effective vocabulary lists?

(2) Do the different types that are removed for the purposes of the AG have an effect on the measurement of lexical sophistication across levels (and by proxy lexical development)?

(3) Which 2000-lemma vocabulary list reveals level differences in lexical sophistication most clearly?

3. METHOD

3.1 Selection of frequency lists for AG

The first list that was selected for AG was the NGSL. This list, released in 2013, is an updated version of the General Service List from 1953 [31]. Unlike many publicly-available word lists, the NGSL is specifically designed with second language learners in mind, and therefore, relevant to Pitt IEP students. To achieve validity, the NGSL is based on a subset of the large Cambridge English Corpus (CEC) which contains two billion words; the subset selected consists of 272 million words, representative of a number of sub-corpora, most notably 38 million words from the Cambridge learner corpus. As a result of this careful corpus composition, the overall coverage of the NGSL exceeds 90% of the CEC texts. The NGSL was also selected due to its public availability in useful Excel file format and clear division of the lemmas into their headwords and inflected forms. In total, for the AG calculations, we used the 2000 highest-frequency lemmas (in keeping with the standard AG formula), as well as an additional 52 basic lemmas from the NGSL supplementary list such as the months of the year and numbers up to one hundred. In the upcoming version 2.0 of the NGSL, these supplementary items will be included in the overall frequency list [5].

The second list was derived from data from PELIC. This corpus contains both written and spoken data that were collected via a web interface and initially stored in a MySQL database. Students may have contributed data from one to three terms, with an average of two terms. For our dataset, we used only the written data from writing classes at the most common levels, levels 3 (intermediate), 4 (upper-intermediate), and 5 (advanced). The written data are 4.2 million tokens from several L1 backgrounds, but primarily Arabic, Chinese, Korean, Spanish, and Japanese learners. The written data were extracted from the MySQL database and analyzed in Python.

To create a high-frequency list from PELIC, which we call the Pitt Service List Level 3 (PSL-3), we used the same 52 supplementary items from the NGSL (for consistency) and added the next most frequent 2000 words in the learners' output at the intermediate level (level 3). When comparing the two lists, the analysis revealed that in terms of identical lemmas, only 1317 of the PSL-3 are found in the NGSL top 2000, with an additional 178 of the PSL-3 in the NGSL top 3000. Words in the PSL-3 that were not in the NGSL top 2000 fell into three broad categories: (i) cultural: e.g., 'camel', 'pyramid', 'spicy', 'tofu', and 'kimchi'; (ii) names: e.g., 'Japan', 'Colombia', 'Pittsburgh'; and (iii) student life: e.g., 'campus', 'admission', 'visa', and 'homework'.

3.2 ETS Comparison-Validation

For comparative purposes, we ran the same AG calculations on a different, but comparable learner corpus: the ETS Corpus of Non-Native Written English (ETS; [2]). This corpus consists of 12,100 English essays written by TOEFL test-takers in 2006-2007. These test-takers have 11 different L1s (many the same as in PELIC), and the texts are divided equally amongst them (1100 per L1). ETS split test takers into proficiency rankings of 'low', 'medium', or 'high'. As such, overall differences in AG lexical sophistication could be measured across proficiency bands.

ETS and PELIC share some similarities since both are learner corpora, contain a variety of L1s, and divide into three proficiency levels. However, they differ in that ETS data were collected under test conditions, whereas PELIC data were collected from day-to-day assignments. Nevertheless, we would expect any patterns found in lexical sophistication in one to be mirrored in the other if the underlying learner-corpus-based frequency lists are generalizable beyond our local context. That is to say, the PELIC-based and NGSL-based AG should equally indicate differences in lexical sophistication on both, despite PELIC and ETS not sharing any of the same learners, tasks, or specific writing prompts.

3.3 PELIC data processing

To preprocess the PELIC data samples for AG analysis, various Python libraries such as pandas, spaCy, and NLTK were used. We filtered out all texts with less than 70 words, following [12], who had a minimum of 66-word texts in their corpus. This process reduced the number of texts from 48,384 to 16,227, but only reduced the token count by 13% from 4,232,746 to 3,736,556. Further filtering of the data was then required as learners in the Pitt IEP revised and re-submitted assignments, often resulting in multiple versions of the same text; the dataset was therefore screened to include only the first version each essay. In addition, within each level and L1 group, there is variance in terms of proficiency and the number of texts and tokens produced. To account for this variation, we calculated average AG scores for individuals to prevent any skewing of data by prolific writers.

Manipulation of the texts was kept to a minimum, and we made a conscious decision to not correct some spelling errors. For example, if a student meant to write 'pot' or 'raw' but due to potential phonological influence on spelling wrote 'port' or 'row', these contextual spelling errors were neither screened nor corrected. However, misspelled tokens were excluded from analysis if they resulted in a non-word (as determined by NLTK's WordNet Synsets as a spellchecker). Such a step was necessary in order to avoid having misspelled basic words like 'thier' register as an advanced type, thereby inflating the AG score. To illustrate the significant effect that misspellings which create non-words can have on lexical sophistication measures, in the ETS data, Arabic low-proficiency texts had an average AG of 1.3 when misspellings were included, whereas this figure dropped to 0.37 when non-word misspellings were excluded from calculation.

Another consideration was advanced-level lexical items found in the writing prompts, which are frequently repeated in student responses. After considering removal of such lexical items from calculations, we ultimately decided to leave them in because the fact that the student 'took up' and used the words in their writing suggests that some learning may have occurred.

Each text was then tokenized using regular expressions. Finally, these tokens were lemmatized, taking the third approach described in section 2. Having completed the above data cleaning process, the resulting data for analysis was comprised of the numbers of texts in Table 1 and individual students in Table 2.

Table 1. Number of texts > 70 words by L1 and level

Level	Arab	Chin	Japan	Korea	Span
3 (Intermediate)	844	307	89	408	116
4 (Upper-Int.)	1659	1001	400	1191	234
5 (Advanced)	1229	851	271	797	184

Table 2. Numbers of students by L1 and level

Level	Arab	Chin	Japan	Korea	Span
3 (Intermediate)	131	48	14	63	13
4 (Upper-Int.)	210	101	39	120	29
5 (Advanced)	141	71	27	86	20

4. RESULTS

4.1 AG measurements of PELIC data

To reiterate, AG is defined as:

$$AG = \frac{Advanced\ Types}{\sqrt{Tokens}}$$

Section 4 describes the results of computing AG using two different high-frequency lists: NGSL and PSL-3. Tables 3 and 4 report the results in that order and the corresponding figures display the mean AG data with standard error bars indicating variability.

Table 3. AG with NGSL on PELIC mean (stdev)

Level	Arab	Chin	Japan	Korea	Span
3	0.63 (0.23)	0.64 (0.23)	0.66 (0.17)	0.77 (0.22)	0.67 (0.15)
4	0.75 (0.25)	0.80 (0.28)	0.83 (0.26)	0.78 (0.21)	0.88 (0.31)
5	0.85 (0.33)	1.06 (0.32)	1.05 (0.29)	0.94 (0.31)	1.03 (0.23)

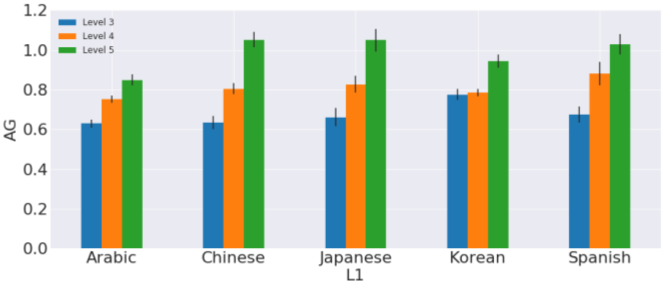


Figure 1. Average AG (using NGSL) on PELIC

Table 4. AG with PSL-3 on PELIC mean (stdev)

Level	Arab	Chin	Japan	Korea	Span
3	0.33 (0.19)	0.31 (0.16)	0.32 (0.19)	0.47 (0.22)	0.33 (0.10)
4	0.57 (0.28)	0.64 (0.31)	0.63 (0.30)	0.59 (0.23)	0.72 (0.39)
5	0.74 (0.37)	0.99 (0.40)	0.94 (0.40)	0.88 (0.37)	0.97 (0.34)

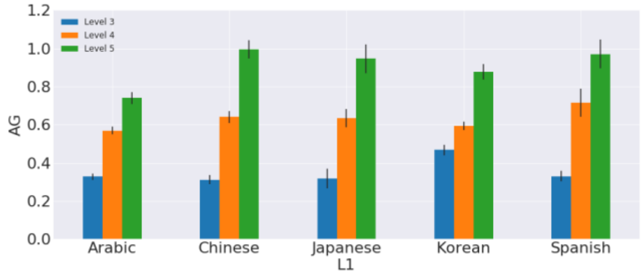


Figure 2. Average AG (using PSL-3) on PELIC

The results in Tables 3 and 4 show that for all L1s, some reliable and consistent group increases are evident in AG as proficiency level increases, regardless of whether NGSL or PSL-3 are used in the AG calculations. Thus, the NGSL means and PSL-3 means distinguish AG among levels. Although standard deviations are high, hand-calculated Confidence Intervals (CI) at the 95% critical value (1.96) show mostly non-overlapping means. This is true for all L1 groups with the exception that the Spanish speakers show an overlap of upper and lower CI for levels 4 and 5 with NGSL. Also noticeable is the difference between levels 3 and 4 for Koreans when using NGSL, as the increase in AG is not significant unlike for the other L1s. However, when PSL-3 is used, this lack of increase is corrected, showing greater increase as would be expected.

However, NGSL and PSL-3 differ in the AG scores that they produce. PSL-3 returns lower AG scores overall, but shows greater range, e.g., approximately 0.31 (Chinese level 3) to 0.99 (Chinese level 5) (a range of 0.67), compared to 0.64 (Chinese level 3) to 1.06 Chinese level 5 (a range of 0.42) for NGSL. The AG scores being lower overall for PSL-3 confirms that PSL-3 includes more words that the learners already know. However, by level 5, AG scores are comparable regardless of the high-frequency list used, indicating that they receive credit for high-frequency words which they later learn. Additionally, with PSL-3, level scores across all L1s appear more distinctly and uniformly segregated: all Level 5 scores regardless of L1 are higher than Level 4 scores. This was not the case with NGSL: the Arabic Level 5 score, for instance, is seen on par with Level 4 scores of other L1s, suggesting (incorrectly) that Arabic Level 5 students are at a similar level of lexical sophistication to, say, Spanish Level 4 students.

In terms of specific L1 differences, there are clear effects for Arabic and Spanish speakers. Overall, Arabic speakers have a lower range and Spanish speakers have a higher range. This lower range in the Arabic speakers' data is manifested across both AG measures, but the upper bound CI for level 5 with PSL-3 was lower than the lower bound CI at level 5 when using NGSL. This result again suggests that PSL-3 is appropriately discounting low-frequency, culture-specific words which learners already know that would otherwise inflate their AG score.

4.2 AG measurements of ETS data

For comparative purposes, we then measured AG in the same way using NGSL and PSL-3, but this time on the ETS corpus. Tables 5 and 6 report the results in that order and the corresponding figures present the mean AG data with standard error bars.

Table 5. AG with NGSL on ETS mean (stdev)

Level	Arab	Chin	Japan	Korea	Span
low	0.34 (0.22)	0.37 (0.26)	0.34 (0.20)	0.38 (0.21)	0.43 (0.23)
medium	0.48 (0.26)	0.58 (0.31)	0.51 (0.20)	0.60 (0.21)	0.55 (0.23)
high	0.82 (0.44)	0.91 (0.42)	0.68 (0.30)	0.83 (0.41)	0.79 (0.38)

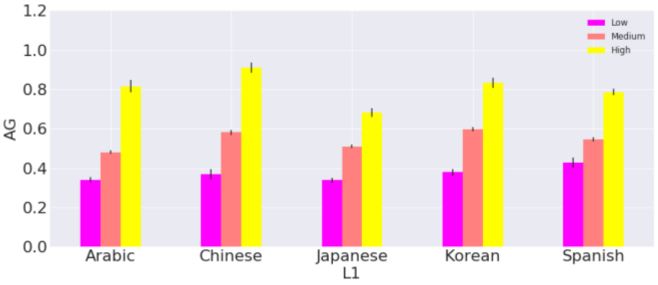


Figure 3. Average AG (using NGSL) on the ETS Corpus

Table 6. AG with PSL-3 on ETS mean (stdev)

Level	Arab	Chin	Japan	Korea	Span
low	0.31 (0.24)	0.35 (0.27)	0.30 (0.22)	0.33 (0.21)	0.44 (0.25)
medium	0.49 (0.32)	0.60 (0.37)	0.548 (0.28)	0.57 (0.31)	0.57 (0.33)
high	0.95 (0.51)	1.02 (0.53)	0.74 (0.38)	0.87 (0.46)	0.93 (0.48)

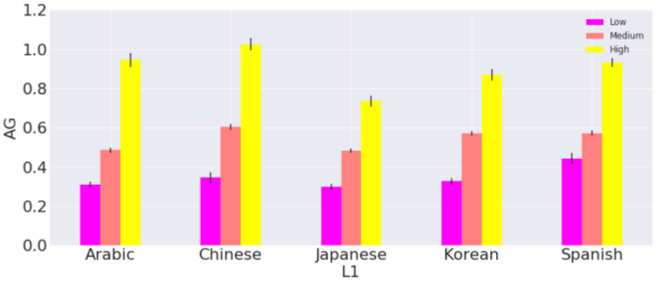


Figure 4. Average AG (using PSL-3) on ETS

These results from the comparison ETS corpus reveal a great deal of consistency in terms of the trends described in 4.1. We acknowledge that the essays in ETS are labelled ‘low’, ‘medium’, and ‘high’, and as such are not strictly comparable to the level system in PELIC. Nevertheless, the AG which was based on PSL-3 appears more effective at showing differences in lexical sophistication than NGSL, as would be expected for learners of different proficiency levels completing an international proficiency exam like TOEFL. This pattern suggests that the findings in 4.1 are not purely specific to the Pitt IEP context, but importantly can be generalized to other learner datasets (though not as effectively as compared to the local context).

5. DISCUSSION

5.1 Differences in frequency lists

To return to our research questions, for question 1, we have demonstrated how data science methods, and specifically natural language processing (NLP) suites such as spaCy and NLTK in Python, can be successfully used to automatically produce vocabulary lists through lemmatization, removal of non-word spelling errors, and token frequency counts.

Regarding research question 2, we showed in answer to question 1 that different frequency lists could be created and deployed and that the choice of corpus affects which high-frequency words are included. In our analysis of our two high-frequency word lists for calculating AG, we found that both NGSL and PSL-3 can show reliable increases as proficiency level increased. These increases in lexical sophistication were detected in both the local learner corpus, PELIC, and the international learner corpus, ETS, validating PSL-3. In addition, the analysis shows that for each L1, AG increases significantly from level to level. (The exception was Spanish-speaking learners from level 4 to 5; this result may be due to low-frequency words being based on Greek and Latin roots which the Spanish speakers control more easily.)

In answer to question 3, we found that the results from the two frequency lists differ in terms of the degree to which AG levels increased with proficiency levels. Overall, the learner-corpus based frequency list yielded more distinct AG differences from level to level, indicative of how we would expect AG to increase with a learner’s overall lexical development over time in an instructed context. Here we acknowledge that the level-by-level data described is cross-sectional, but it can serve as a proxy for longitudinal growth; in future work, hierarchical linear modeling (HLM) will be used to statistically confirm this claim. (HLM is appropriate as not all learners provide a data point at each level, but this statistical approach allows one to compensate for this issue, e.g., [29]) Instead, at present we are restricting the analysis to the calculation of mean scores with confidence intervals, thereby allowing us to provide descriptive evidence of differences in AG when different lists are used.

Our explanation for this finding is that learners may already know and control some less frequent NGSL words at a low-intermediate stage due to cultural background but may not know some words that occur in the 2000 most frequent words in a native speaker corpus. This knowledge inflates AG at lower proficiency levels. In other words, when measuring lexical development against a native-speaker corpus, learners incorrectly get credit for less frequent words that they already know (items not in the frequency list from their culture or educational context), but do not get credit for words that they learn when these more frequent items become known to them. Thus, native speaker-based frequency measures may present a less nuanced picture of the L2 productive lexicon. The learner-corpus frequency list provides more differentiated AG scores, resulting in a more clearly stratified picture of learner knowledge across levels, and by extension, predicted longitudinal growth.

5.2 Importance of data science tools

These observations were made possible by data analysis of very large numbers of texts and tokens. To our knowledge, data mining analysis of a corpus of learner data of this nature, with a variety of L1s and a similarity of educational experience in an IEP, has not been reported before in the literature. Although a subset of the

PELIC spoken data was hand-coded and made public (see, e.g., http://alpha.talkbank.org/data-cmdi/talkbank-data/SLABank/English/Vercellotti/) and several articles published since [20, 29, 30], the potential for far greater insights into development in an IEP are possible from analysis of the whole dataset. Therefore, the ability to analyze a learner corpus of this size is an important step forward in more precise characterization of 'academic readiness', which is an issue in IEP programs that prepare international students for academic programs [15].

5.3 Limitations and L1 effects

We acknowledge that there are limitations at this early stage of exploration. For example, we have yet to determine the exact effect of task prompts or the most reliable manner of lemmatizing our own high-frequency lists with open-source tools. Another area for investigation is the degree to which specific L1 characteristics affect their AG measurements. For example, it has been documented in PELIC that Arabic learners tend to misspell more than other L1s [14]. By excluding all non-word misspellings, Arabic learners may not receive credit for words they may know in all senses except for the spelling. This finding is important as Arabic speakers' knowledge of the L2 may be underestimated and thus put them at a disadvantage in standardized proficiency tests, which are the gateway to quality higher education programs.

6. CONCLUSION

This paper used data mining techniques to provide evidence that AG measures of lexical sophistication will provide more accurate descriptive data if they are based on learner corpora (e.g., PSL-3) rather than frequency lists based on native speaker corpora (e.g., NGSL). The work presented here shows that mining a large dataset that has been collected from an L2 population can provide more fine-grained insight into level differences, and by implication development, than data that are less closely associated with the learners. This research is also a good example of how applied linguists and data scientists can collaborate to provide results from very large datasets, combining linguistic theory with data analysis.

As a next step, we plan to conduct further analysis and comparisons using other corpora and word lists as the basis for calculations. The Cambridge English: Preliminary and Preliminary for Schools Vocabulary List (PET; [7]) which is based on the Cambridge Learner Corpus, a subset of the CEC, is an obvious choice. As this list is intended to indicate words that a learner at CEFR level B1 should possess, it would seem a well-suited comparison to PSL-3. It may be that an ideal frequency list would consist of a combination of a local (like PSL-3) and a global (like PET) list in order estimate learner knowledge and their lexical needs.

We will also explore additional quantitative validation metrics, such as comparing AG scores with various frequency lists to general proficiency measures. We would also like to know whether culture-specific words such as 'camel', 'pyramid', 'tofu' and 'spicy' should be counted for all L1s. It is natural that Arab-speaking learners already know 'camel', but perhaps not Japanese learners, who are more likely to be familiar with 'tofu'. Would L1 specific versions of PSL-3 change the outcomes for each L1 and would materials writers for each L1 context find such L1-specific lists useful?

Overall, this research has the potential to inform numerous areas of language teaching. For materials writers, curriculum planners, and teachers, there is great value in having easy access to a valid list of level- and context-appropriate vocabulary on which to base classroom lessons. For testing services such as ETS or other institutions interested in automated assessment of proficiency levels, such lists can improve the reliability and validity of measurements related to lexical sophistication, and by extension, overall lexical development. Finally, in terms of research in this field, transparent and theoretically-motivated list selections allow for improved comparisons and reproducibility across studies. We therefore see this paper as a step in closing the gap between educational data mining research, classroom instruction, and assessment in the ESL industry.

7. ACKNOWLEDGMENTS

We would like to thank the teachers and students of the English Language Institute at the University of Pittsburgh and grants from the National Science Foundation via the Pittsburgh Science of Learning Center (http://learnlab.org), funded award number SBE-0836012. (Previously NSF award number SBE-0354420.)

8. REFERENCES

[1] Bauer, L. and Nation, P. 1993. Word families. *International Journal of Lexicography*, 6, 4, 253-79.

[2] Blanchard, D., et al. 2014. ETS Corpus of Non-Native Written English LDC2014T06. Philadelphia: Linguistic Data Consortium, 2014.

[3] Brezina, V. and Gablasova, D. 2015. Is There a Core General Vocabulary? Introducing the *New General Service List*. *Applied Linguistics*, 36, 1, 1, 1–22. DOI: https://doi.org/10.1093/applin/amt018

[4] Browne, C., Culligan, B. and Phillips, J. 2013. The New General Service List. Retrieved from http://www.newgeneralservicelist.org.

[5] Browne, C. 2014. A New General Service List: The Better Mousetrap We've Been Looking For? *Vocabulary Learning and Instruction*, 3, 2, 1-10.

[6] Bulté, B. and Housen, A. 2014. Conceptualizing and measuring short-term changes in L2 writing complexity. *Journal of Second Language Writing* 26, 42-65. DOI: http://dx.doi.org/10.1016/j.jslw.2014.09.005

[7] Cambridge English: Preliminary and Preliminary for Schools Vocabulary List. 2012. Retrieved from http://www.cambridgeenglish.org/images/84669-pet-vocabulary-list.pdf

[8] Cobb, T. 2016. Numbers or numerology? A response to Nation (2014) and McQuillan (2016). *Reading in a Foreign Language* 28, 2, 299-304.

[9] Daller, H., van Hout, R. and Treffers-Daller, J. 2003. Lexical Richness in the Spontaneous Speech of Bilinguals. *Applied Linguistics* 24, 2 (Jun. 2003), 197-222. DOI: https://doi.org/10.1093/applin/24.2.197

[10] Daller, H. and Phelan, D. 2007. What is in a teachers' mind? In Daller, Milton, Treffers-Daller (Eds.) *Modelling and Assessing Vocabulary Knowledge*, (234-244). Cambridge University Press, Cambridge.

[11] Daller, M., Turlik, J., and Weir, I. 2013. Vocabulary acquisition and the learning curve. In S. Jarvis & M. Daller (Eds.), *Vocabulary Knowledge: Human ratings and*

automated measures, 185-218). John Benjamins, Amsterdam.

[12] Daller, H. and Xue, H. 2007. Lexical richness and the oral proficiency of Chinese EFL students. In H. Daller, J. Milton, and J. Treffers-Daller (Eds.), *Modelling and assessing vocabulary knowledge* (150-164). Cambridge University Press, New York.

[13] Davies, M. 2008-. The Corpus of Contemporary American English (COCA): 560 million words, 1990-present. Available online at https://corpus.byu.edu/coca/.

[14] Dunlap, S. 2012. *Orthographic quality in English as a second language.* (PhD), University of Pittsburgh, Pittsburgh, PA.

[15] Hoekje, B.J., & Stevens, S.G. 2017. Creating a culturally inclusive campus: A guide to supporting international students. Routledge, New York.

[16] Gibson, E. and Schütze, C.T. 1999. Disambiguation preferences in noun phrase conjunction do not mirror corpus frequency. *Journal of Memory and Language,* 40, 263-279.

[17] Jarvis, S. 2013. Defining and measuring lexical diversity. In S. Jarvis & M. Daller (Eds.), *Vocabulary knowledge: human ratings and automated measures,* 13-44. John Benjamins, Amsterdam.

[18] Juffs, A. 1998. The acquisition of semantics-syntax correspondences and verb frequencies in ESL materials. *Language Teaching Research,* 2, 93-123.

[19] Juffs, A. (in press). Lexical development in the writing of English Language Program Students. In R. M. DeKeyser and G.P. Botana (Eds.), *Reconciling pedagogical demands with pedagogical applicability*. John Benjamins, Amsterdam.

[20] Li, N., and Juffs, A. 2015. The influence of moraic structure on English L2 syllable final consonants. P. *2014 Annual Meeting on Phonology*. DOI: http://dx.doi.org/10.3765/amp.v2i0.3767

[21] Malvern, D., Richards, B.J., Chipere, N. and Durán, P. 2004. *Lexical diversity and language development.* Palgrave, Basingstoke.

[22] Nation, P. and Waring, R. 1997. Vocabulary Learning Strategies. In N. Schmitt and M. McCarthy (Eds.) *Vocabulary: Description, Acquisition and Pedagogy,* 6-19. Cambridge University Press, Cambridge.

[23] Nation, I.S.P. 2001. *Learning vocabulary in another language*. Cambridge University Press, Cambridge.

[24] Schmitt, N. 2010. Researching Vocabulary. A Vocabulary Research Manual. Palgrave Macmillan, Basingstoke.

[25] Schmitt, N. and Schmitt, D. 2014. A reassessment of frequency and vocabulary size in L2 vocabulary teaching. *Language Teaching,* 47, 4, 484–503. DOI:10.1017/S0261444812000018

[26] Schmitt, D. 2016. Beyond Caveat Emptor: Applying Validity Criteria to Word Lists. In Proceedings of Vocab@TOKYO: Current Trends in Vocabulary Studies. September 12-14, 2016, 17.

[27] Tidball, F. And Treffers-Daller, J. 2008. Analysing lexical richness in French learner language: what frequency lists and teacher judgements can tell us about basic and advanced words. Journal of French Language Studies 18, 3, 299-313. DOI: https://doi.org/10.1017/S0959269508003463.

[28] van Hout, R. and Vermeer, A. 2007. Comparing measures of lexical richness. In H. Daller, J. Milton, and J. Treffers-Daller (Eds.), *Modelling and Assessing Vocabulary Knowledge*, 93-115. Cambridge University Press, Cambridge.

[29] Vercellotti, M.L. 2017. The development of complexity, accuracy and fluency in second language performance. *Applied Linguistics*, 38, 90-111. *Doi.org/10.1093/applin/amv002*

[30] Vercellotti, M.L., & Packer, J. 2016. Shifting structural complexity: The production of clause types in speeches given by English for academic purposes students. *Journal of English for Academic Purposes,* 22, 179-190. DOI: dx.doi.org/10.1016/j.jeap.2016.04.004

[31] West, M. 1953. A General Service List of English Words. London: Longman, Green and Co.

Prediction of Academic Achievement Based on Digital Campus

Zheng Wang
Key Laboratory of Universal Wireless Communications for Ministry of Education
Beijing University of Posts and Telecommunications
Beijing,China
616016323@qq.com

Xinning Zhu
Key Laboratory of Universal Wireless Communications for Ministry of Education
Beijing University of Posts and Telecommunications
Beijing,China
zhuxn@bupt.edu.cn

Junfei Huang
Institute of network technology,Beijing University of Posts and Telecommunications
Beijing University of Posts and Telecommunications
Beijing,China
1465422103@qq.com

Xiang Li
Key Laboratory of Universal Wireless Communications for Ministry of Education
Beijing University of Posts and Telecommunications
Beijing,China
674904403@qq.com

Yang Ji
Key Laboratory of Universal Wireless Communications for Ministry of Education
Beijing University of Posts and Telecommunications
Beijing,China
jiyang@bupt.edu.cn

ABSTRACT

Academic achievement of a student in college always has a far-reaching impact on his further development. With the rise of the ubiquitous sensing technology, students' digital footprints in campus can be collected to gain insights into their daily behaviours and predict their academic achievements. In this paper, we propose a framework named AAP-EDM (Academic Achievement Prediction via Educational Data Mining) to predict students' academic achievements based on the influencing factors we have discovered. Multi-source heterogeneous data including Wi-Fi detection records, usage of smartcards, usage of campus network, is aggregated firstly. Then, instead of the self-reported features or traditional academic assessments like test scores, we extract features reflecting students' behavioural patterns. Specially, we define **DOH** (Degree of Hardworking) to improve the performance of the classifier. Finally, we analyze the features extracted and apply supervised learning methods to predict their academic achievements. Experiments are conducted on real-world data from 528 college students in one faculty, and the classification accuracy can be up to 88%.

Keywords

Digital footprints, academic achievement prediction, multi-source data merging, supervised learning, behavioural pattern

1. INTRODUCTION

Predicting students' academic achievements is one of the most popular applications in Educational Data Mining. One research predicted students' academic achievements by analyzing students' static information such as gender, character, eating habits and place of residence.[2]. Authors used predictive modeling methods to identify at-risk students in a course using standards-based grading.[5]. Authors found that students' achievements were best inferred from their social ties through modified smartphones.[4]. Researchers demonstrated the impact of students' psychology in predicting their academic achievements using examination scores, information processing abilities as features [3]. Under the circumstance of online learning, researchers predicted 145 students' academic achievements utilizing their online learning activities and online discussion forums [7, 8]. There are also authors who used passive sensing data and self-reports from students' smartphones and proposed a model based on linear regression with lasso regularization to predict **GPA** [9].

Our study is conducted to make up for the two shortcomings in the previous studies. On the one hand, compared with standard academic assessments or personal static information, students' daily behaviours which can be monitored anytime can reflect their states of living and learning more sensitively and timely. Past research has shown that students' academic achievements have relationships with their daily behaviours [9]. We inspect students' behaviours by analyzing their trajectories, class schedule, campus network usage and smartcard usage. On the other hand, our study is conducted based on a complete passive detection system with no active participation of students which facilitates continual studies of a larger scale [6, 10]. It is important to mention that we care about the privacy protection very much and all of students' information involved in the study is anonymous.

In this paper, we propose a framework named **AAP-EDM** (Academic achievement prediction via educational data mining) to analyze data generated from digital campus in order to predict students' academic achievements. The framework contains mainly three main modules. Multi-source heterogeneous data merging is the first. After that, we extract features such as wake-up time, duration of stay in the dormitory, and class attendance. We discovered the potential influencing factors of academic achievements through ANOVA F-test and correlation coefficients analysis. Furthermore, we defined the feature **DOH** (Degree of Hardworking) to consider the features we have extracted comprehensively. Then, we formalized the prediction as a binary classification problem to identify students at risk and choose the best solution from multiple classification algorithms consisting of SVM, Logistic Regression,Naive Bayes and Decision Tree. Finally, we evaluated the proposed framework over a real-world dataset involving 528 undergraduates, and found that the classification accuracy can be up to 88%.

Our main contributions in this paper are listed below:

(1) We predicted students' academic achievements utilizing students' daily life behaviour data rather than using academic assessments such as test scores. The high accuracy rate indicates that students' academic achievements have strong relationships with their daily behaviours.

(2) We extracted abundant features which can describe students daily life in detail and also define the **DOH** which improves the performance of classifiers.

(3) In order to explore students' behaviour patterns extensively, we came up with methods to fuse the multi-source heterogeneous data of college students. Our research can be easily expanded to much larger scale.

2. PROBLEM FORMULATION

Our raw data consists of four components. First, students' usage of campus network is monitored in real time. Then when students use their smartcards on campus such as when having breakfast and going shopping, their behaviours will also be captured. Moreover, through the Wi-Fi monitors we deployed in the entrance of particular places in the campus, Wi-Fi packets from students' smartphones with Wi-Fi enabled can be captured when they pass by the monitors without connecting to the network. Besides the three parts above, we have static data including students' class schedules and academic achievements. We will introduce the data set in detail in the next section. Based on the data, our target is to extract features of students and train models utilizing supervised learning algorithms to predict academic achievements.

Formally, given the input matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$ where M represents the total number of students and N is the number of features which will be introduced later and the academic achievements labels matrix $\mathbf{Y} \in \mathbb{R}^{M \times 1}$, our target is to learn the function which satisfies $\mathbf{Y} = f(\mathbf{X})$. Note that the labels in our study are either 0 or 1 where 0 represents good performers and 1 represents students at risk.

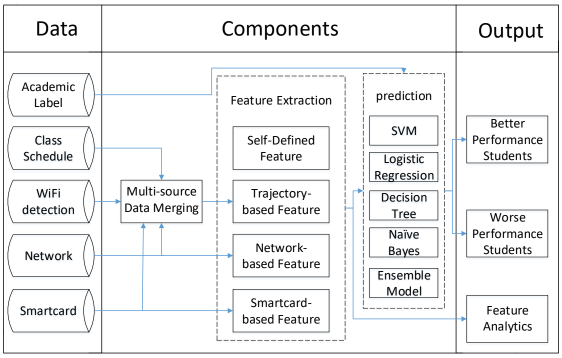


Figure 1: Overview of the framework

Table 1: Data format of Wi-Fi detection records.

MAC Address	Time	RSSI	Location
38BC******91	20160301 12:20:23	-70	Canteen #1

3. METHODS

In this section, we will introduce our framework **AAP-EDM** in detail. The framework mainly contains multi-source heterogeneous data merging, feature extraction and academic achievement prediction which is illustrated in figure 1.

3.1 Multi-source Heterogeneous Data Merging

3.1.1 Raw Data Set

The raw data set contains Wi-Fi detection records, usage of campus network, usage of smartcards, class schedules and also students' academic achievements.

Through deploying Wi-Fi monitors at entrances of locations such as dormitories, canteens and teaching buildings, it is possible to detect smartphones' MAC addresses, providing a coarse-grained location trace for students who enter the coverage area of Wi-Fi monitors which is shown in Table 1.

Students' information of using campus network is shown in Table 2. Specifically, the locations where students access the network (building-level) can be inferred from the ”IP Address”, and the ”Network Traffic” describes the traffic between login time and logout time in MBs, which includes uplink traffic and downlink traffic.

The information of students' devices while connected to the campus network is shown in Table 3. In the table, the ”Device Type” can help us distinguish mobile devices from PC and the ”Time” is recorded in days but not seconds compared with Table 1.

Table 4 demonstrates the usage of smartcards. The ”Consumption Type” includes ”Repast”, ”Shopping”, ”Bathing”,

Table 2: Data format of usage of campus network

Anonymous ID	IP Address (Location)	Login/ logout Time	Network Traffic
E416**B2ED	10.210.**.***	20160301 08:00:00/ 20160301 09:00:00	200

Table 3: Data format of device information

MAC Address	IP Address (Location)	Device Type	Time
38BC******91	10.210.**.**	Mobile	20160301

”Network cost” and so on. Note that the consumption type will reveal the location where students consume with their smartcards.

Other than the data mentioned above, in this paper we also utilize students' class schedules to analyze students' class attendance and utilize students' academic achievements to train the classification model.

3.1.2 Trajectory Generation

We arranged the usage of campus network, the usage of smartcards and Wi-Fi detection records in chronological order to form students' semantic trajectories. In particular, we consider students to stay in the specific location during the periods between the login time and logout time according to campus network records, until records are captured in other locations. The semantic trajectories are shown in Table 5.

3.2 Feature Extraction

3.2.1 Trajectory Features

Daily wake-up time: Wake-up time can reflect the degree of diligence to a certain extent which is calculated as the first time in a day when a student logs in to the network in his dormitory.

Daily time of return to dormitory: Returning to dormitories at a later time in the evening usually means longer periods students spend in the classrooms or the library. We regard the last time in a day when a student logs in to the network in his dormitory as the time of return to dormitory.

Daily duration spent in the dormitory: Dormitories are usually not appropriate places for studying. We can estimate the duration of time spent in dormitory according to the time that students enter and leave the dormitory. Specially, only the time between 06:30 and 23:30 is under consideration.

Table 4: Data format of usage of smartcard

Anonymous ID	Time	Cost	Consumption Type (Location)
E416**B2ED	20160301 08:00:00	5.0	Repast

Table 5: Example of a semantic trajectory in one day

Id	Time	Location
1	07:30:00	Dormitory #13
2	07:33:14	Canteen #1
3	08:21:52	Teaching Building #3
4	11:49:39	Canteen #2
5	12:50:58	Dormitory #13
6	18:03:58	Canteen #2
7	18:35:34	Dormitory #13
8	20:39:16	Teaching Building #2
9	22:08:56	Super Market
10	22:15:15	Dormitory #13

Class Attendance: Given the daily trajectory $\{p_0 \rightarrow p_1 \rightarrow ... \rightarrow p_n\}$ where $p_n = (loc, time)$, the start time t_s and the end time t_e of the course according to class schedules, we will judge whether a student attends the class. Considering that students must appear in the classrooms and shouldn't have any records in other irrelevant places during the class, we propose the method according to two conditions. Eq.1 ensures that students have no records except in classrooms during the class periods. Eq.2 ensures that students are indeed in the classrooms.

$$\{p | t_s + \Delta t < time < t_e - \Delta t, loc \neq classroom\} = \emptyset \quad (1)$$

$$\{p | t_s - \Delta t < time < t_e + \Delta t, loc = classroom\} \neq \emptyset \quad (2)$$

Days outside of campus: Students who have no digital footprints in one day will be considered as not on campus. Students' academic achievements are supposed to be affected if they are often not on campus.

3.2.2 Network Features

Daily Network Traffic in Dormitory: We sum up the network traffic that students upload and download in their dormitories. Compared with dormitories, the network traffic in teaching buildings is less, so we don't take this part into consideration.

Network Cost: Students don't need to pay for the campus network until their used traffic exceeds the upper limit of every month. The upper limit of network traffic is almost enough for normal usage, so students who exceed the limit may spend too much time on the internet accessing online videos or online games. We calculate the total network charges of each student.

Network top up Frequency: When the balance of students' network accounts is zero, students should recharge for continual usage.

Daily Network Traffic Peak: Daily network traffic peak is demonstrated as $L = \{l_0, l_1, ..., l_{23}\}$ where l_n represents an hour in a day and takes value of 0 or 1 shown in Eq.3 where $traffic_n$ is the traffic during the n_{th} hour and the $average$ is the average traffic per hour in one day.

$$l_n = \begin{cases} 1, & traffic_n >= average \\ 0, & traffic_n < average \end{cases} \quad (3)$$

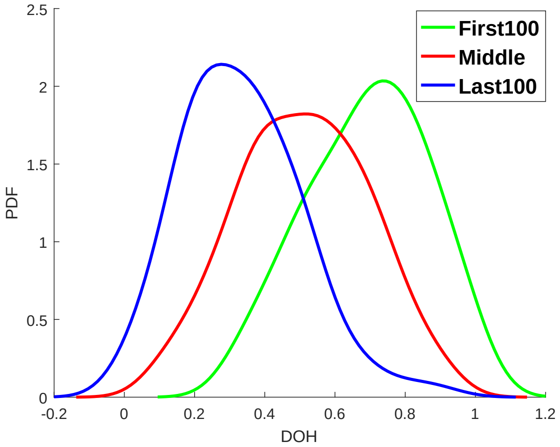


Figure 2: Probability density function of DOH

3.2.3 *Smartcard Features*

Students' consumption patterns are captured according to the usage of smartcards. In the campus, students will use their smartcards when having meals in the canteens, shopping in the supermarket and taking a shower in the bathhouse. Cumulatively, we calculate students' daily costs and frequency of consumption of breakfast, shopping and bathing.

3.2.4 *Self-defined Features*

In order to obtain a comprehensive evaluation of all features extracted above, we calculated the score of each feature for each student Eq.4. $Corr(X_k)$ is the Pearson correlation coefficient between the k_{th} feature X_k and student's academic achievements which is shown in Table 6. Note that the academic achievements are in the form of rankings when calculating the Pearson correlation coefficient. $Rank(x_n)$ means the ranking of the student u_n' features among N students. For example, there are three students (u_1, u_2, u_3), and their i_{th} feature (class attendances) are $(0.8, 0.5, 0.6)$, we have $Score_i^1 = 1, Score_i^2 = 0.66, Score_i^3 = 0.33$ because $Corr(X_i) < 0$ according to Table 6.

Then we defined the degree of hardworking(**DOH**) utilizing the feature scores Eq.5 where K is the count of all features we have extracted. We plot the probability density function of **DOH** (Min-Max normalized) of three groups of students separated by their rankings of academic achievements as shown in Figure 2. From the figure we can find that the distributions of DOH are similar to the normal distribution and the averages are approximately 0.2, 0.5 and 0.8. The apparent distinction among three groups proves that our defined feature is a strong factor for prediction. Essentially the DOH is the weighed mean of feature scores and the weighs are the correlation coefficients. Besides DOH , self-defined features also include other statistics characteristics of feature scores such as average and median.

$$Score_k^n = \begin{cases} (N - Rank(x_n))/N, & Corr(X_k) > 0 \\ Rank(x_n)/N, & Corr(X_k) < 0 \end{cases} \tag{4}$$

$$DOH^n = \sum_{k=1}^{K} (|Corr(X_k)| * Score_k^n) \tag{5}$$

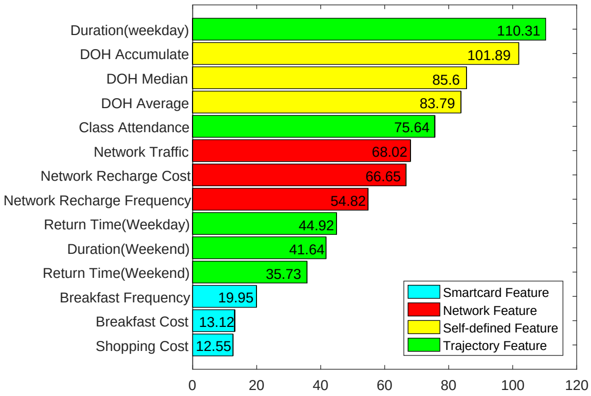


Figure 3: ANOVA F-test for binary classification

3.3 Academic Achievement Prediction

We separate the whole semester into four periods, the first three periods last for four weeks respectively and the last one lasts for six weeks. We calculate the mean of daily features respectively in four periods due to the fact that students' behaviours may change along with the whole semester and generate different impacts on their academic achievements. Moreover, it is necessary to distinguish weekdays and weekends in each period for different behavioural patterns.

The academic achievement prediction is essentially a binary classification problem which can be used in academic precaution. For that the values of features vary greatly, in order to increase the speed of gradient descent and the accuracy of classifiers, we limited all the feature values to the range of 0 to 1 using Min-Max normalization. We have 100 students who performed the worst according to their school reports to be positive labels and other 428 students to be negative labels. The dataset is split into training set and test set according to the ratio of 7:3.

There might be relevancies among features which will decrease the performance of classifiers. For example, students who spend long time surfing the campus network can possibly bear high network charges. In this work, we implement the state-of-the-art methods, Principal Component Analysis, to solve this problem.

We trained various classification models such as Logistic Regression, Support Vector Machine, Naive Bayes and Decision Tree using cross-validation and evaluated on the test set. Moreover, we implemented the voting classifier to combine conceptually different machine learning classifiers and use a majority vote to predict the class labels.

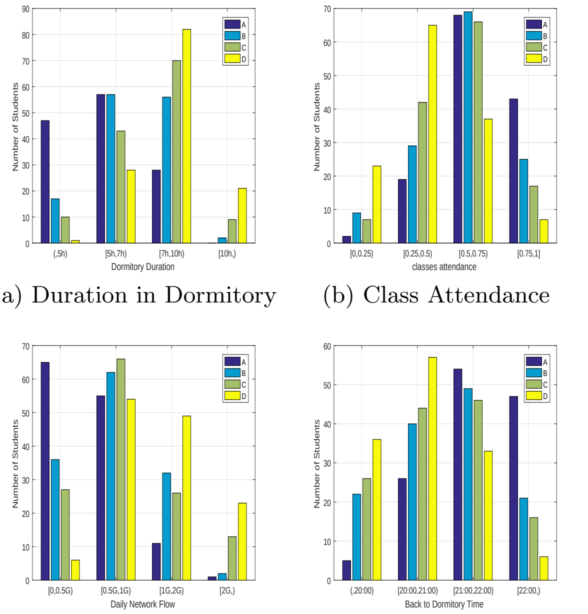
4. EXPERIMENTAL RESULTS

4.1 Experimental Data

We collect 1673706 records totally of 528 undergraduates in their third year from 19 classes in one faculty. The period we selected lasted for a complete semester of 140 days from Feb. 29^{th}, 2016 to Jul. 17^{th}, 2016. The academic achievements

Table 6: Correlation Coefficient and P-value

Feature	Correlation coefficient	P-value
Class attendance	-0.430	3.39e-25
Time spent in dormitory(Weekday)	0.565	7.71e-46
Time spent in dormitory(Weekend)	0.411	5.84e-23
Time of return to dormitory(Weekday)	-0.394	4.22e-21
Time of return to dormitory(Weekend)	-0.348	1.60e-16
Wake-up time(Weekday)	0.222	2.69e-7
Wake-up time(Weekend)	0.204	2e-6
Shopping cost	0.215	6.09e-7
Breakfast Frequency	-0.337	1.9e-15
Breakfast cost	-0.266	5.55e-10
Days out of campus	0.068	0.117
Network traffic	0.406	2.11e-22
Network cost	0.362	8.3e-18
Network top up frequency	0.361	1.02e-17
Feature score average	-0.551	3.3e-43
Feature score median	-0.547	1.64e-42
DOH	-0.561	3.84e-45



(a) Duration in Dormitory (b) Class Attendance

(c) Daily Network Traffic (d) Returning Dormitory Time

Figure 4: Features statistics among different grades

are the weighted average scores of all courses in a semester which includes quizzes, midterns and finals.

What we should emphasize is that our experiment was completely conducted under an anonymous situation. In our experiment, students' IDs which reveal the true identities were mapped to anonymous IDs.

4.2 Feature Analytics

It is meaningful for educators to find out how students' behaviours influence the academic achievements. We performed correlation coefficient analysis and ANOVA F-test to compare different features' contributions to students' study. The correlation coefficients are shown in Table 6. Note that time spent in dormitory on weekdays reaches the highest value of 0.565 with smallest P-value which is a novel finding. Our self-defined features also reach high correlation coefficients. Many previous studies have shown that class attendance is a significant and positive predictor of academic achievements which is also true in our study. Specifically, the self-defined features indicate high correlation coeffiients which is also proved in the ANOVA F-values for binary classification shown in Figure 3. Thus it can been seen that our proposed method for new features is effective which will improve the performance of the prediction. Other than the self-defined features, the overall F-values of network features are relatively high while the smartcard features are slightly irrelevant. Note that the wake-up time and the days leaving campus which don't achieve sufficient significance ($p \geq 0.001$) are omitted in the Figure 3.

To observe the differences of behaviours among students in detail, we display the distributions of four features which are highly relative with academic achievements in Figure 4. We divide all the students into four groups in the order of their academy achievements. Group A represents the best performers and group D represents the worst performers.

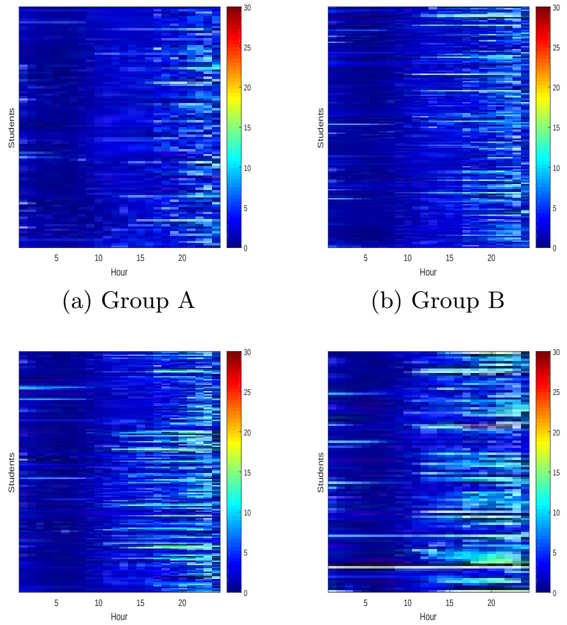
As we can see in subgraph Figure 4a, more than 70% students of group A spend less than 7 hours in dormitories. On the contrary, most students in group C and D stay in dormitories for longer than 7 hours, some even staying for more than 10 hours. In subgraph Figure 4b, we find that class attendance is mainly distributed from 0.5 to 0.75 except group D in which more than 60% students' attendance is less than 0.5. Nearly 90% students of group A have a high attendance rate. Whether class attendance has influence on academic achievements is controversial.[9, 1] We discover that it is a relatively strong factor in our research. Daily network traffic is shown in subgraph Figure 4c, it is obvious that more than 90% students spend less than 1 GB traffic daily in group A. Bad performers may spend more time for online gaming and

Table 7: Classification Results

Model	Class0 Precision	Class0 Recall	Class0 F1-score	Class1 Precision	Class1 Recall	Class1 F1-score	Accuracy
SVM	0.92	0.86	0.89	0.55	0.69	0.61	0.82
SVM(PCA)	0.87	**0.97**	**0.92**	0.78	0.44	0.56	0.86
LR	0.92	0.77	0.84	0.45	0.75	0.56	0.77
LR(PCA)	0.88	**0.97**	**0.92**	**0.79**	0.47	0.59	0.87
NB	0.92	0.71	0.80	0.39	0.75	0.52	0.72
NB(PCA)	0.87	0.96	0.91	0.72	0.41	0.52	0.85
DT(PCA)	0.91	0.93	**0.92**	0.73	0.69	0.71	0.87
SVM+LR(PCA)	**0.94**	0.91	**0.92**	0.69	**0.75**	**0.72**	**0.88**

movies which results in more network traffic. Subgraph Figure 4d shows students' time of return to dormitory. The left two groups of data tend to show an ascending trend while the right ones show a descending trend which depict that most students of group A and B come back to dormitories after 21:00 and are therefore more diligent.

Figure 5 shows the distribution of students' daily network rush hours in one month. The horizontal axis represents the 24 hours in one day. The vertical axis represents students in the specific group according to academic achievements. Each student is represented by a row vector ($\boldsymbol{v} \in \mathbb{R}^{24}$) accumulated in one month according to Eq.3. The color bar shows the numbers in vectors which are between 0 and 30 (30 days in one month). Therefore, the brighter areas mean students always spend more time online during the specific periods. From the figure we can see, students of group A and B have a shorter span of rush hours and they always login the network near to 22:00 after they come back from classrooms, while rush hours of students of group C and D last for a longer time from about 15:00 to 23:00.



(a) Group A

(b) Group B

(c) Group C

(d) Group D

Figure 5: Daily network rush hours

4.3 Results of Prediction

In our research the prediction task is an unbalanced classification problem. According to students' academic achievements, the dataset is composed of 428 good performers (negative samples) and 100 bad performers (positive samples). We conducted four different supervised learning algorithms consisting of Support Vector Machine, Logistic Regression, Decision Tree and Naive Bayes. The highest classification accuracy can be up to 88%. However it is not convincing enough for unbalanced classification problems to just inspect the classification accuracy. In this paper, we used precision, recall and F1-score to evaluate the performance of our models. The average classification results of 10-Fold cross validation are shown in Table 7. Specially we ensemble the Support Vector Machine and Logistic Regression through voting classifier and realize the highest accuracy 88%. The principle of the voting classifier is that the students are classified as negative samples when the two classifiers conflict with each other.

5. CONCLUSIONS

In this paper, we predicted that students' academic achievements to identify students who perform worse in their study based on our proposed framework **AAP-EDM**. Firstly, multisource heterogeneous data is merged to generate semantic trajectories. Then we extracted features consisting of trajectory features, network features and smartcard features. Furthermore, self-defined features are proposed to explore features comprehensively. At last, we have evaluated the framework through multiple classification models using students' real world data. The results show that our proposed framework is feasible and meaningful for educational supervision and warning. Our research provides promising approaches to transform the collage education from traditional descriptive analytics to predictive analytics. We will improve our framework through further research and concentrate on realizing the prescriptive analytics in college education.

6. ACKNOWLEDGMENTS

This paper is supported by "the Fundamental Research Funds for the Central Universities",(No.2018XKRK03).

7. REFERENCES

[1] J. Brocato. How much does coming to class matter? some evidence of class attendance and grade performance. *Educational Research Quarterly*,

19(3):2–6, 1989.

[2] T. Devasia, V. T P, and V. Hegde. Prediction of students performance using educational data mining. In *Data Mining and Advanced Computing (SAPIENCE), International Conference on*, 2016.

[3] R. R. Halde, A. Deshpande, and A. Mahajan. Psychology assisted prediction of academic performance using machine learning. In *IEEE International Conference On Recent Trends In Electronics Information Communication Technology*, 2016.

[4] V. Kassarnig, A. Bjerre-Nielsen, E. Mones, S. Lehmann, and D. Dreyer Lassen. Academic performance and behavioral patterns. Website, 2017. https://arxiv.org/abs/1706.09245.

[5] F. Marbouti, H. A. Diefes-Dux, and K. Madhavan. Models for early prediction of at-risk students in a course using standards-based grading. *Computers & Education*, 103:1–15, 2016.

[6] A. Musa and J. Eriksson. Tracking unmodified smartphones using wi-fi monitors. In *SenSys '12 Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 281–294, 2012.

[7] A. Pardo, F. Han, and R. A. Ellis. Combining university student self-regulated learning indicators and engagement with online learning events to predict academic performance. *IEEE Transactions on Learning Technologies*, 10(1):82–92, 2016.

[8] C. Romero, M. LÃşpez, J. Luna, and S. Ventura. Predicting students' final performance from participation in on-line discussion forums. *Computers & Education*, pages 458–472, 2013.

[9] R. Wang, G. Harari, P. Hao, X. Zhou, and C. T. Smartgpa: How smartphones can assess and predict academic performance of college students. In *UBICOMP '15,OSAKA, JAPAN*, 2015.

[10] S. Zhao, Z. Zhao, Y. Zhao, R. Huang, S. Li, and G. Pan. Discovering peopleâĂŹs life patterns from anonymized wifi scanlists. In *Ubiquitous Intelligence and Computing, 2014 IEEE 11th Intl Conf on and IEEE 11th Intl Conf on and Autonomic and Trusted Computing, and IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UTC-ATC-ScalCom)*, 2014.

A Hybrid Multi-Criteria approach using a Genetic Algorithm for Recommending Courses to University Students

Aurora Esteban
University of Cordoba
Dept. of Computer Science
14071, Cordoba, Spain
i32estoa@uco.es

Amelia Zafra
University of Cordoba
Dept. of Computer Science
14071, Cordoba, Spain
azafra@uco.es

Cristóbal Romero
University of Cordoba
Dept. of Computer Science
14071, Cordoba, Spain
cromero@uco.es

ABSTRACT

This paper describes a multiple criteria approach based on a hybrid method of Collaborative Filtering (CF) and Content-Based Filtering (CBF) for discovering the most relevant criteria which could affect the elective course recommendation for university students. In order to determine which factors are the most important, it is proposed a genetic algorithm which automatically discovers the importance of the different criteria assigning weights to each one of them. We have carried out an in-depth study using a real data set with more than 1700 ratings of Computer Science graduates at University of Cordoba. We have used different proposals and different weights for each criterion in order to discover what is the combination of multiple criteria which provides better results.

Keywords

Educational recommender system, Course recommendation; Hybrid Multi-Criteria Approach; Genetic Algorithm

1. INTRODUCTION

Course recommendation is nowadays an interesting and increasing research line. Specifically, course recommendation for university studies can be viewed as an important educational data mining task [13]. This is a important problem because university studies normally provide a number of elective courses which students have to choose to complete their studies. This decision may not be trivial for students, which usually don't have enough information and get overwhelmed by the amount of available options. Recommender Systems (RS) appear as essential tools capable of helping students choosing relevant elective courses in their curriculum according to different criteria such as their individual ratings, preferences, interests, needs, performance, etc [6]. Although there are some studies which work with hybrid RS approaches [2, 9] and multiple criteria approaches [10, 16], these works are fairly and are not focused on studying the influence of the different factors in the recommendation process. This work presents a preliminary study to determine which are the most relevant criteria to provide better course recommendations for university students. These criteria include both information that describes the students (such as their ratings, their grades and their branch) and information that describes the courses (such as their competences, their theoretical and practical contents, the professors that teach it and their subject area). In order to determine which factors are the most important to achieve better course recommendations, a force brute search and a Genetic Algorithm (GA) are proposed. GA automatically discovers the importance of the different criteria assigning weights to each one of them. Then, these weights are incorporated to the recommendation process in order to make a final suggestion to students. In order to study the advantages and limitations of using different criteria, a real dataset which includes information from the Computer Science degree at University of Cordoba is used.

The rest of this paper is organized as follows. An overview of related work is specified in Section 2. The proposed methodology is presented in Section 3. The description of the experimental study is described in Section 4. Finally, conclusions and future work are presented in Section 5.

2. RELATED WORK

In the past few years, RSs have been thoroughly applied to course recommendation using multiple criteria. One of the first applications of multi-criteria matrix factorization for course rating predictions is explored in [15]. Later, Vialardi et al. [16] proposed multi-criteria techniques for predicting students' grades as a classification problem and Parameswaran et al. [12] explored the application of restrictions to recommendations using multiple criteria. Also, other techniques can be found in course recommendation, for instance, ontology-based approaches [5, 18], neural networks [7] or bio-inspired algorithms with proposals such as ant-colony optimization [14] and artificial immune systems [2]. Most of them based only in students' grades. From other perspective, the study of the importance of the specific moment in which the courses are taken has been studied based on students' grades using Markov chains [8] as well as applying multiple criteria [17]. More recently, both the competences provided to students and their relevance in their recommendation [4, 1] and the application of semantic analysis [11] has been adressed.

In conclusion, even though several techniques have been developed for course recommendation, most of them are mainly focused on the students' performance and do not use further criteria. Even when some other criteria are used, a study to determine each criterion influence on the quality of recommendations is not carried out. In this paper, we propose a multi-criteria approach for discovering the most relevant criteria which could affect the course recommendation. Our approach combines student information (known as Collaborative Filtering, CF) with domain-specific information (known as Content-Based Filtering, CBF).

3. PROPOSED METHODOLOGY

This section describes the proposed methodology (Figure 1). First, a description and analysis of data set is presented. Then, the recommendation approaches and the criteria used in each one of them are detailed. Finally, the evaluation methodology is addressed.

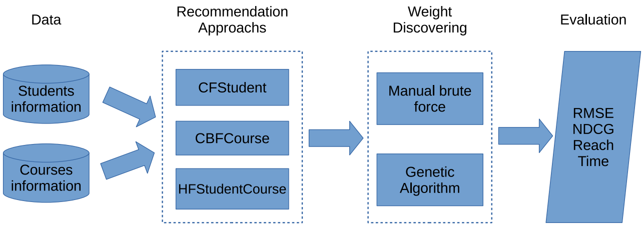


Figure 1: Methodology overview.

3.1 Data description and preparation

This work has been developed using real information gathered from the degree of Computer Science at University of Cordoba, Spain. This includes information about students and courses.

3.1.1 Student information

Student information was obtained by means of surveys which students filled in their last academic year. The factors obtained for each student are represented in the following way (see Figure 2):

- A rating of the overall students' satisfaction for each course. It is a integer value from 0 to 5 if the course is taken or it is empty otherwise.
- The grade obtained by students on each course. It is a decimal value in the range $[0, 10]$ if the course is taken or an empty value otherwise.
- The branch selected by students for specializing in a particular computer science area. Concretely, Computer Science degree offers three branches: Computation, Computer Engineering or Software Engineering. The chosen of the student will be represented as a numeric identifier (from 1 to 3).

In total, more than 1700 ratings along with their corresponding grades were obtained for the 63 courses included in Computer Science degree in University of Cordoba, Spain. The data was gathered over a period of two years (2016-2017).

To avoid global effects in the grades and ratings *subtractive normalization* [15] is applied. This normalization subtracts a combination of the student and course mean to the original value.

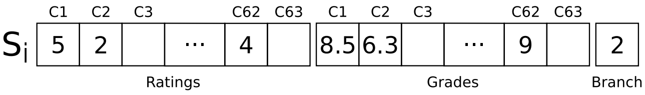


Figure 2: Student information.

3.1.2 Course information

Course information was obtained from the University official degree web page[1]. The factors selected for each course are represented in the following way (see Figure 3):

- The professors involved in the course, represented as a vector with an index for each professor in the degree. Its value is 1 if the professor is involved in this course or 0 otherwise.
- The competences or skills that the course provides, represented as a vector with an index for each competence in the degree. Its value is 1 if it is provided by the course or 0 otherwise.
- The subject area to which the course belongs, represented as a numeric identifier. Eight subject areas are considered in the degree (integer value from 1 to 8).
- The contents of the course, represented as a frequency vector of keywords obtained by text mining/preprocessing the theoretical and practical contenst of the course.

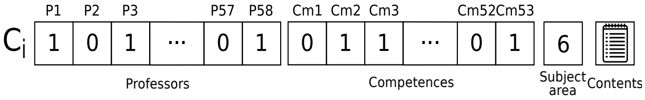


Figure 3: Course information.

3.2 Recommendation Approaches

Three different recommendation approaches are appproposed to evaluate the influence of students and courses criteria.

3.2.1 Collaborative Filtering using student information - CFStudent

This proposal follows a CF approach where each student is represented using different factors, such as, the ratings vector, the grades vector and the branch. For the courses not taken by a student, the estimated preferences are obtained based on the neighborhood built using a similarity function.

For each pair of students, i and j, the similarity measure designed considers on one hand the ratings ($R_{i,j}$) and the grades ($G_{i,j}$). These similarities are calculated using metrics like Pearson or Spearman correlation coefficients and euclidean or taxicab distances. On the other hand, it is considered the branch similarity ($B_{i,j}$). This similarity is computed considering whether it is equal or not. All these measures are mapped into the $[0, 1]$ interval and the final similarity measure is computed as a parametric linear combination of the three factors:

$$\begin{aligned} D_{U_{i,j}} &= \alpha \cdot R_{i,j} + \beta \cdot G_{i,j} + \gamma \cdot B_{i,j} \\ &where\ \alpha + \beta + \gamma = 1 \end{aligned} \tag{1}$$

The significance of each criterion can be studied according to the weight (α, β or γ) assigned to each criterion. Finally, the final preference for student i and course j, $U_{i,j}$, is calculated using the parametrized similarity measure (equation 1).

[1] http://www.uco.es/eps/node/619

3.2.2 Content-Based Filtering using course information - CBFCourse

This proposal follows a CBF approach where each course is represented as a series of features, such as, the subject area, the contents, the professors and the competences. In this approach, the course recommendations for a student are based on the estimated ratings of the most similar courses to those that they have already taken.

For each pair of courses, i and j, the similarity measure is designed attending to the following criteria: their professors ($P_{i,j}$), their competences ($Cm_{i,j}$) and their respective subject area ($S_{i,j}$). These similarities are computed considering whether they are shared or not. Also, it is considered a semantic analysis based on their contents ($Cn_{i,j}$). All measures are mapped into the $[0, 1]$ interval. The final similarity measure is computed as a parametric linear combination of these four factors:

$$\begin{aligned} D_{C_{i,j}} =& \alpha \cdot P_{i,j} + \beta \cdot Cm_{i,j} + \gamma \cdot S_{i,j} + \delta \cdot Cn_{i,j} \\ & where\ \alpha + \beta + \gamma + \delta = 1 \end{aligned} \tag{2}$$

The significance of each criterion can be studied according to the weight (α, β, γ or δ) assigned to each factor (equation 2). To compute similarities based on professors and competences, a boolean data based approach is followed. Thus, similarity metrics like Jaccard index or the log-likelihood function can be used.

Similarity based on course contents is stored as keywords obtained by preprocessing the theoretical and practical contents described in the course official guide. Therefore, semantic similarity is applied to each pair of courses in the following manner:

1. First, the documents are indexed: a custom text parser has been implemented based on the language (in our case, Spanish) and it is used a set of stop words adapted to the domain. As a result, for each document, a list of tokens is obtained along with their frequency as well as the number of times that each one appear in the document.

2. For each pair of courses, i, j, a set B is created as the union of the tokens of both courses. For each course, a vector $\vec{i}$ or $\vec{j}$ is built with as many elements as there are in B, represented as n. This vector contains the frequency of each token. Finally, each vector is normalized using the l_1 norm, thus it is obtained the relative frequencies to each pair of courses.

3. Cosine similarity is applied to both frequency vectors in order to integrate the course content criterion into the similarity measure between courses.

$$cos(\theta) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\| \cdot \|\vec{j}\|} = \frac{\sum_{k=1}^{n} i_k j_k}{\sqrt{\sum_{k=1}^{n} i_k^2}\sqrt{\sum_{k=1}^{n} j_k^2}} \tag{3}$$

Finally, the final preference for student i and course j, $C_{i,j}$, is calculated using the parametrized similarity measure (equation 3).

3.2.3 Hybrid Filtering using student and course information - HFStudentCourse

To avoid some of the problems of CF and CBF systems, a hybrid approach is proposed. The course preference estimation for each student and course is obtained using a linear aggregation of the estimated preference based on student information described in section 3.2.1 and the estimated preference based on course information described in section 3.2.2. Both estimations are decimal numbers in range from 1 to 5, so they are combined with certain weights α and β to provide a final preference estimation also in this range. Hence, for the student i and the course j, the preference estimations according to CFStudent ($U_{i,j}$) and to CBFCourse ($C_{i,j}$) are combined into a final estimation ($p_{i,j}$):

$$\begin{gathered} p_{i,j} = \alpha \cdot U_{i,j} + \beta \cdot C_{i,j} \\ where\ \alpha + \beta = 1 \end{gathered} \tag{4}$$

This hybrid approach implies two different configuration levels. A first level where student and course information are used separately to obtain two preference estimations. Then, a second one where it is configured the relevance of each criterion in the final recommendation.

3.3 Weights selection

Two different ways to select the weights have been used in order to configure each recommendation approach.

3.3.1 Exhaustive search

A brute-force search or exhaustive search has been used to find the best weights. This method consists on systematically enumerating all possible weight configurations and checking which configuration obtains the best results. In our case the different weights studied have been considered as decimal numbers between 0 and 1 with increases of 0.1. This type of search has been used for the CFStudent and CBFCourse approaches due to the fact that they do not have a very high number of weight combinations.

3.3.2 Genetic Algorithm

A GA has been also used to automatically discover the best weights. This has only been used for the HFStudentCourse approach due to the larger number of parameters and, therefore, more potential configurations. Its purpose is to find the optimal weights of the different criteria concerning student and course information, as well as the weights of the final linear aggregation to obtain the final preference estimation. The more relevant factors achieve higher weights and the less relevant ones, the lowest values. The main components of the used GA algorithm are:

- The chromosome is defined with integer values to represent the weight of each factor. The integer value of each gene is ranged from 0 to 10 and it would represent to the percentage in the range of $[0, 1]$. A total of 9 weights have to be assigned in this approach, three weights assigned to student information, four weights assigned to course information, and finally, two weights to determine the relevance in the final estimation considering CFStudent and CBFCourse approaches.

  The previous study of exhaustive search allows assigning restrictions to assign specific weights to particular criterion to reduce the search space. Thus, three different parameters are optimized deducing the rest of the problem restrictions.

- The individual fitness function is the Root-Mean-Squared Error (RMSE) of the recommendation when using the weight configuration given by the chromosome.
- The genetic operators are single point crossover and a random mutation which changes the value of one gene in a possible value in the fixed range.
- Parent selection is done by binary tournament.

3.4 Evaluation Metrics

There are several standpoints from which a RS performance can be evaluated [3]. In this proposal four metrics have been selected attending to accuracy, relevance or capability of making recommendations.

3.4.1 Root-Mean-Squared Error

The Root-Mean-Squared Error (RMSE) is used to measure the accuracy of the recommendations. This measure is suitable for the prediction of ratings and it tends to penalize larger errors more severely than other metrics. If $p_{i,j}$ is the predicted rating for student i over course j, and $v_{i,j}$ is the true rating and $K = \{(i,j)\}$ is the set of hidden student-course ratings, then the RMSE whose purpose is to minimize is defined as:

$$RMSE = \sqrt{\frac{\sum_{(i,j)\in K}(p_{i,j} - v_{i,j})^2}{\#K}} \tag{5}$$

3.4.2 Normalized Discount Cumulative Gain

Attending to Information Retrieval (IR), normalized Discount Cumulative Gain (nDCG) is used as measure of ranking quality.

$$nDCG = \frac{DCG}{IDCG} \tag{6}$$

DCG at a particular rank position p, if rel_i is the graded relevance of the result at position i, is defined as:

$$DCG = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)} \tag{7}$$

Normalization is given by the division by the Ideal DCG at position p (IDCG).

3.4.3 Reach

CF is based on similarities between students. Depending on the criteria used, some outlier users exist for which no satisfactory similarities are found, and so no recommendation can be made for these users. This behavior will be measured by the reach of the RS whose purpose is to maximize. If $K = \{(i,j)\}$ is the set of hidden student-course ratings and $p_{i,j}$ is the predicted rating, reach is defined as:

$$Reach = \frac{\#K - \sum_{(i,j)\in K} p_{i,j}}{\#K} \; \forall \; p_{i,j} = \emptyset \tag{8}$$

3.4.4 Time

The execution time of each approach is also important. The mean execution time is analyzed once each model has been learned. It is calculated the time that each approach takes on building the recommendation ranking for a user. It is important to mention that our testing platform is a personal computer with Ubuntu 16.04 64-bit as operative system, a Intel Core i5-3317U processor and 12 GiB RAM memory, and our recommender runs under the Java Virtual Machine.

4. EXPERIMENTAL WORK

We have carried out two experimental studies. Firstly, we show the criteria weight optimization and then the comparative study between the different approaches developed. As mentioned in section 3.1, the dataset used comes from real ratings and grades gathered from students of University of Cordoba.

The different RS approaches have been implemented using Apache Mahout[2] and the GA has been developed using the JCLEC library [3].

It is important to notice that in order to guarantee a greater robustness in the results and so they can be generalized to an independent data set, a 10-fold cross validation has been used. We have stratified students' data according to the volume of received ratings on each course [3]. In essence, a portion of ratings from each student will be taken away to train the RSs with the remaining ratings. Then, data are divided into ten partitions, and each partition in turn is used as a test set. In this way, the obtained results in the different evaluation measures represent the average values of the test data set for each fold considered. The advantages of the cross-validation approach are to allow the use of more data in ranking algorithms, and to take into account the effect of training set variation.

4.1 Criteria Weight optimization

The main objective of this first experimental study is to find the optimal weights for each criterion used in the proposed RSs. Thus, it is evaluated the influence of the weights in the course recommendation.

Firstly, an initial experimental study is carried out to configure some common parameters, such as, the similarity metrics, where the Jaccard index and the log-likelihood function have been evaluated for categorical values, and the Pearson correlation and the euclidean and taxicab distances have been evaluated for numerical values. Also, neighborhood size has been evaluated with the values of 5, 10 and 15 in the case of CFStudent and HFStudentCourse. The final selected configuration according to this study is shown in Table 1. This configuration of common parameters will be used by our three RS approaches.

Next, the weight optimization of each criterion used in CFStudent and CBFCourse approaches is carried out by means of exhaustive search. Figure 4 shows the evolution of the average RMSE and its standard deviation for the CFStudent approach, varying the weight assigned to the ratings

[2] https://mahout.apache.org/

[3] http://jclec.sourceforge.net/

Table 1: Similarity measure and neighborhood size.

Similarity by ratings	
CFStudent	Euclidean distance
HFStudentCourse	Euclidean distance
Similarity by grades	
CFStudent	Taxicab distance
HFStudentCourse	Euclidean distance
Similarity by professors	
CBFCourse	Log-likelihood function
HFStudentCourse	Log-likelihood function
Similarity by competences	
CBFCourse	Jaccard index
HFStudentCourse	Jaccard index
Neighborhood size	
CFStudent	10
HFStudentCourse	15

and grades criteria, maintaining fixed and with 0.1 value the weight for branch factor. According to these values, it can be affirmed that ratings criterion is considered more relevant than grades criterion. Thus, higher weights for the ratings factor provide better recommendations (lower RMSE values). However, if only the ratings criterion is used (assigning a weight of 1.0 and 0.0 for the other criteria), it can be appreciated that the RMSE value is worse than when using the rest of criteria with lower values. Concretely, the best weight configuration is shown in Table 2. In this manner, although with lower relevance, it is also important to consider these criteria (grade and branch) in order to improve the results.

In the case of the CBFCourse approach, Figure 5 shows the RMSE evolution, attending to its average and its standard deviation, varying the weights of content and professor criteria (considered the two factors more representative in this approach) and maintaining fixed and with minimum values (that is, 0.1 value) the weight for competences and subject area factors.

The results demonstrate that the lowest RMSE values are obtained when both factors use averaged weights. Specifically, the best configuration gives a lower weight to the competences and subject area factors. Then, the content factor is also representative but its weight is slightly lower than the weight assigned to the professor criterion. The best configuration is shown in Table 2.

Finally, in the case of HFStudentCourse, because of the increase in complexity, nine different factors have to be optimized, the weights have been estimated using the GA proposed whose main parameters are population size: 100, number of generations: 500, mutation probability: 0.2 and crossover probability: 0.9. For this approach, the Figure 6 shows the evolution of the best weight configuration obtained by the GA in different generations showing the RMSE mean values and the obtained weights of the most relevant factors in the two hybridized proposals. Note that there are some secondary criteria whose weights aren't reflected in the graph since they were pre-fixed. Concretely, the branch criterion in CFStudent approach with a specific weight of 0.1, and subject area and competences with a weight of 0.1 for each one of them in CBFCourse approach. For the best configuration obtained in the last generation, the weights are not exactly the same values than the other approaches separately, but the tendency is similar: the ratings criterion obtains higher weight values than other criteria of student information and the professor obtains slightly higher weight values with respect to content criterion. Moreover, the weights to determine the importance that should be given to the results of CFStudent approach and CBFCourse approach for combining them and obtaining a final recommendation show that the best combination is obtained by maintaining a balance between both criteria. In our case, the best configuration has a weight of 0.6 for CFStudent approach, 0.4 for CBFCourse approach and the rest of weights shown in Table 2.

.

Table 2: The best weight configurations.

Criterion	CFStudent	CBFCourse	HF[1]
Ratings	0.8	–	0.6
Grades	0.1	–	0.3
Branch	0.1	–	0.1
Professors	–	0.4	0.5
Subject area	–	0.1	0.1
Competences	–	0.1	0.1
Content	–	0.4	0.3
CFStudent	–	–	0.6
CBFCourse	–	–	0.4

[1] HFStudentCourse

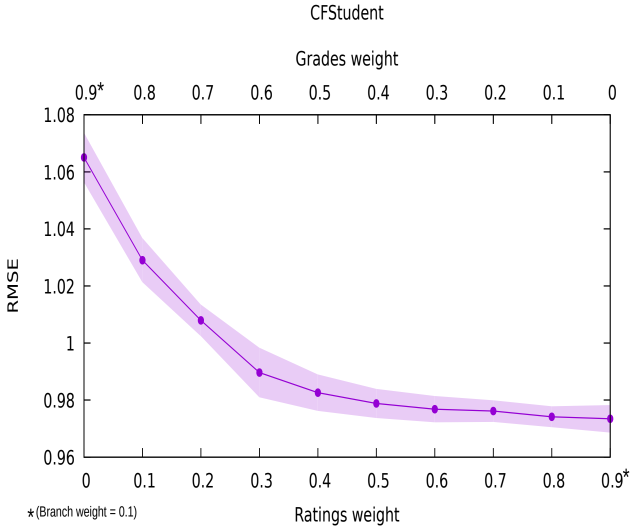


Figure 4: Weighted criteria of CFStudent approach.

Starting obtaining the best configuration for each approach, the following conclusions can be obtained:

- The weight assigned to each criterion indicates that the most important criterion for student information is the ratings. In the case of course information, course contents and professors' criteria take the lead.
- The similarity measures for ratings and grades based on distance predominate over the ones based on lin-

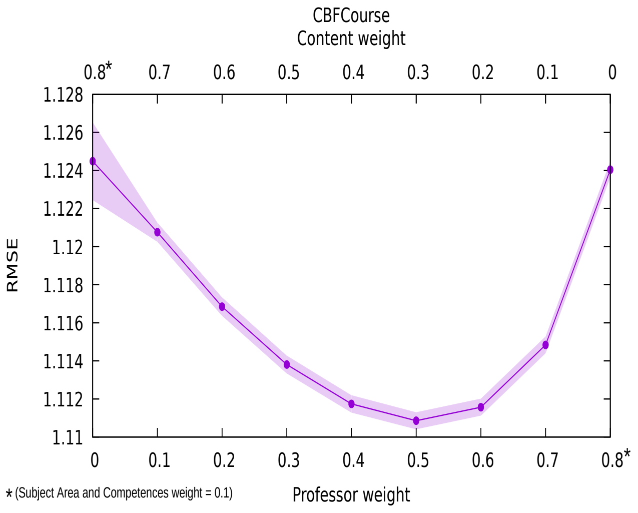


Figure 5: Weighted criteria of CBFCourse approach.

ear relationships. Moreover, the optimal neighborhood size grows with the number of criteria used.

- The best weight configurations for CFStudent and CBFCourse are not exactly the same considering the proposals separately or combined in hybrid approach, but the tendency is maintained. Moreover, the hybrid approach assigns a balanced weight to both proposals to obtain the final recommendation. Thus, both approaches are considered necessary to obtain the best recommendations.

4.2 Comparison of the different approaches

This second experimental study compares the results obtained by the best configurations of the previous approaches. We have used an estimation of the ratings (RMSE) as well as the others of the evaluation measures (nDCG, reach and execution time) described in section 3.4.

Table 3: Comparative evaluation between RS.

	RMSE	nDCG	Reach	Time
CFStudent	0.96628	0.7980	96.48%	1.53s
CBFCourse	1.11187	0.2768	99.36%	1.81s
HFStudentCourse	1.04150	0.8955	100%	2.05s

As we can see in the results shown in Table 3 for the RMSE, a better score is obtained when more information about the student and less about the course is used. Nonetheless, course information provides certain advantages, such as increasing the number of ratings capable of estimating (*reach*) or a more diverse set of solutions (*nDCG*), which can translate into a better proficiency in making relevant recommendations. As expected, as the amount of information considered is increased, the time taken in finding the recommendations for a student is also increased. It is then concluded that, regarding RMSE optimization, the best approach consists in using just the student information, improving as multiple criteria based on it are introduced, although explicit ratings still have the most weight. However, this approach has important flaws, as it is the capability of obtaining ratings for all users, because of outlier students for whom it is difficult to find an appropriate enough neighborhood. This shortfall is overcome when information about courses is introduced. It is practically guaranteed that similarities between courses will be found, so the reach score increases significantly.

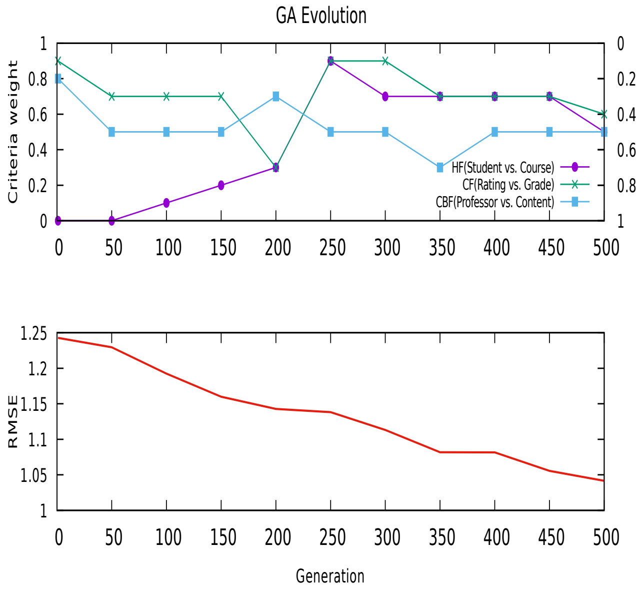


Figure 6: Weighted criteria of HFStudentCourse approach.

5. CONCLUSIONS

In this paper several proposals based on CF, CBF and hybrid RS approaches combining multiple criteria have been proposed for the task of elective courses recommendation in university studies. The results confirm that the overall rating that a student gives to a course is the most reliable information source, but when it is complemented with other criteria about the own student or the course then the estimation accuracy can improve it. This work opens a promising line of research geared towards both data enhancement, by applying the RS to a larger volume of students and majors and study transferability, and broadening the used models beyond CF. The application of a GA to search for optimal configurations also has potential, especially on the modeling of chromosomes capable of containing information apart from the weights of the criteria. As future work, we want to evaluate weights to all criteria (including the criteria that we have pre-fixed). Moreover, other parameters such as, size of neighbour and similarity metrics also could be optimized. Finally, it is also important to indicate that our proposed approach could be also applied to other related educational domains such as recommendation of massive open online courses (MOOCs) with only adapting the used factors.

6. ACKNOWLEDGMENTS

Authors gratefully acknowledge the financial subsidy provided by Spanish Ministry of Science and Technology TIN2017-83445-P.

7. REFERENCES

[1] B. Bakhshinategh, G. Spanakis, O. Zaiane, and S. ElAtia. A course recommender system based on graduating attributes, January 2017.

[2] P.-C. Chang, C.-H. Lin, and M.-H. Chen. A hybrid course recommendation system by integrating collaborative filtering and artificial immune systems. *Algorithms*, 9(3), 2016.

[3] A. Gunawardana and G. Shani. A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research*, 10:2935–2962, 2009.

[4] J.-w. Han, J.-c. Jo, H.-s. Ji, and H.-s. Lim. A collaborative recommender system for learning courses considering the relevance of a learner's learning skills. *Cluster Computing-The Journal of Networks Software Tools and Applications*, 19(4):2273–2284, 2016.

[5] C.-Y. Huang, R.-C. Chen, L.-S. Chen, and Ieee. Course-recommender system based on ontology. In *Proceedings of 2013 International Conference on Machine Learning and Cybernetics*, pages 1168–1173, 2013.

[6] T.-N. Huynh-Lv, N. Huu-Hoa, and T.-N. Nguyen. Methods for building course recommendation systems. In M. LeNguyen, L. S. Vinh, L. T. Bui, V. G. Nguyen, Y. S. Ong, and K. Hirata, editors, *Eighth International Conference on Knowledge and Systems Engineering*, pages 163–168, 2016.

[7] A. A. Kardan, H. Sadeghi, S. S. Ghidary, and M. R. F. Sani. Prediction of student course selection in online higher education institutes using neural network. In *Computers & Education*, volume 65, pages 1–11, 2013.

[8] E. S. Khorasani, Z. Zhenge, and J. Champaign. A markov chain collaborative filtering model for course enrollment recommendations. In *IEEE International Conference on Big Data (Big Data)*, pages 3484–3490. IEEE, 2016.

[9] C. Kim, N. Choi, Y. Heo, J. Sin, and . On the development of a course recommender system: A hybrid filtering approach. *Entrue Journal of Information Technology*, 14(2):71–82, 2015.

[10] F. Le Roux, E. Ranjeet, V. Ghai, Y. Gao, J. Lu, and A. T. PRess. A course recommender system using multiple criteria decision making method. In *Proceedings of the International Conference on Intelligent Systems and Knowledge Engineering*, Advances in Intelligent Systems Research, 2007.

[11] H. Ma, X. Wang, J. Hou, Y. Lu, and Ieee. Course recommendation based on semantic similarity analysis. In *Conference Proceedings of 2017 3rd Ieee International Conference on Control Science and Systems Engineering (Iccsse)*, pages 638–641, 2017.

[12] A. Parameswaran, P. Venetis, and H. Garcia-Molina. Recommendation systems with complex constraints: A course recommendation perspective. *ACM Transactions on Information Systems*, 29(4):20:1–20:33, 2011.

[13] C. Romero and S. Ventura. Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 3(1):12–27, 2013.

[14] J. Sobecki and J. M. Tomczak. Student courses recommendation using ant colony optimization. In N. T. Nguyen, M. T. Le, and J. Swiatek, editors, *Intelligent Information and Database Systems, Pt Ii, Proceedings*, volume 5991 of *Lecture Notes in Artificial Intelligence*, pages 124–133, 2010.

[15] S. Spiegel. *A Hybrid Approach to Recommender Systems based on Matrix Factorization.* Thesis, Technical University Berlin, 2009.

[16] C. Vialardi, J. Chue, J. Pablo Peche, G. Alvarado, B. Vinatea, J. Estrella, and A. Ortigosa. A data mining approach to guide students through the enrollment process based on academic performance. *User Modeling and User-Adapted Interaction*, 21(1-2):217–248, 2011.

[17] R. Wang. *Sequence-based Approaches to Course Recommender Systems.* Thesis, University of Alberta, 2017.

[18] L. Zhuhadar, O. Nasraoui, R. Wyatt, and E. Romero. Multi-model ontology-based hybrid recommender system in e-learning domain. In R. BaezaYates, B. Berendt, E. Bertino, E. P. Lim, and G. Pasi, editors, *IEEE/WIC/ACM International Joint Conferences on Web Intelligence*, pages 91–95, 2009.

Tracking Behavioral Patterns among Students in an Online Educational System

Stephan Lorenzen
University of Copenhagen
lorenzen@di.ku.dk

Niklas Hjuler
University of Copenhagen
hjuler@di.ku.dk

Stephen Alstrup
University of Copenhagen
s.alstrup@di.ku.dk

ABSTRACT

Analysis of log data generated by online educational systems is an essential task to better the educational systems and increase our understanding of how students learn. In this study we investigate previously unseen data from Clio Online, the largest provider of digital learning content for primary schools in Denmark. We consider data for 14,810 students with 3 million sessions in the period 2015-2017. We analyze student activity in periods of one week. By using non-negative matrix factorization techniques, we obtain soft clusterings, revealing dependencies among time of day, subject, activity type, activity complexity (measured by Bloom's taxonomy), and performance. Furthermore, our method allows for tracking behavioral changes of individual students over time, as well as general behavioral changes in the educational system. Based on the results, we give suggestions for behavioral changes, in order to optimize the learning experience and improve performance.

Keywords

Student clustering, Non-negative matrix factorization, Educational Systems

1. INTRODUCTION + RELATED WORK

How students behave in educational systems is an important topic in educational data mining. Knowledge of this behavior in an educational system can help us understand how students learn, and help guide the development for optimal learning based on actual use. This behaviour can be understood both through an explicit study [5], or as in this paper through the automatically generated log data of the system.

The analysis of log data is usually done as an unsupervised clustering of students [2, 3, 4, 7]. A popular approach is to extract action sequences and transform them into an aggregated representation using Markov models [4, 7]. The Markov chains can then be clustered by different methods. Klingler et al. did student modeling with the use of explicit Markov chains and the clustering with different distance measures defined on the Markov chains [7]. Hansen et al. assumed the actions sequences to be generated by a mixture of Markov chains and used an heuristic algorithm to find the generating Markov chains [4]. Gelman et al. used non-negative matrix factorization to find clusters for different measures of activity aggregated in weekly periods during a MOOC course. These clusters are then matched from week to week by cosine similarity.

Our work is similar to Gelman et al. [3] in that we also use *Non-negative Matrix Factorization* (NMF) to make a soft clustering at the student level in a given time period, however our clustering is only made once, and we are looking at primary school data over a vastly longer period of time, (2 years compared to 14 weeks).

Our soft clustering by non-negative matrix factorization is based on log data from Clio Online.[1] Clio Online is the largest provider of digital learning for all subjects in the Danish primary school (except mathematics), having 90% of all primary schools in Denmark as customers.

Using NMF, we assume that the set of features chosen can be represented by a set of fewer underlying behaviors. These underlying behaviours would each be represented by a cluster in the non-negative matrix factorization. Each student will then get a number for each cluster in each time period representing how much of that underlying behavior he has shown in the given time period. Non-negativity gives the behaviors an additive structure, which is more natural than showing a negative amount of a given behavior. We reason that the soft clustering will show both the behaviors of individual students, as well as how the behaviors change over time, both individually and on a system-wide level.

In this paper, we will consider two main questions: a) how does student activity in the system affect performance, and b) how does student activity distribute between different levels of Bloom's taxonomy in different subjects. Both questions are important in regards to optimizing learning; the first in relation to performance, the latter in relation to utilization of all taxonomy levels.

[1] This data is proprietary and not publicly available.

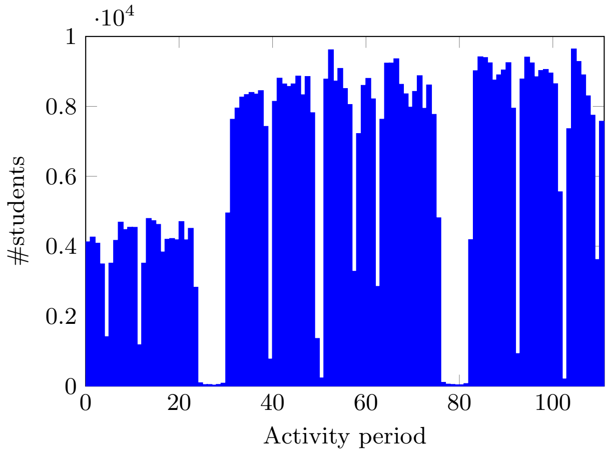


Figure 1: Number of students active in each period. Note that period 0 starts on 2015-01-08, while period 111 ends on 2017-03-01. The drops in activity occur due to vacation in Danish primary school, with the two large drops around periods 25 and 79 being due to the summer vacation.

2. EXPERIMENTAL SETUP

This section describes our experimental setup and methods. We start by describing our data and how it is preprocessed, and then move on to describing our clustering method.

2.1 Data Preprocessing

As mentioned, we consider log data generated in the Danish online educational system Clio Online. The system is used in Danish primary schools and contains learning objects across all Danish subjects (except mathematics), for instance texts, videos, sound clips and exercises. Furthermore, the system includes a large number of quizzes, used for evaluating students. Students may use the system for self study, but they may also be assigned homework by their teacher. Our data covers 14,810 students.

The raw data consists of logs detailing page accesses for individual students in the system. For quizzes, the final score (between 0 and 1) and total time spent for the quiz is also available. In our preprocessing, we combine these log entries to *sessions*. Two consecutive entries are considered in the same session, if they have the same subject, and their timestamps differ by less than some threshold. For our study, we choose this threshold to be 600 seconds, based on recommendations from Clio Online, who have a deeper knowledge of the content and flow of the system (e.g. expected time per page). Furthermore, quizzes are considered separate sessions. A total of 3 million sessions is obtained in this way.

With the sessions defined, we consider student activity in *activity periods*, with a length of one week. The data spans a total of 112 activity periods, starting January 2015 and ending in March 2017. For each activity period, we add an entry for a student, if the student is active (accesses the system) within that period. The entry for the given student contains all sessions for that student, which starts within the activity period. We end up with approximately 677,000 student entries across the 112 periods. Figure 1 shows the active number of students in each period. Note the drop in active students around periods 25 and 79; these drops in activity occur due to summer vacation.

The final step of data preprocessing is the feature extraction. For each activity period, a set of activity/performance related features are extracted. The features are chosen so as to answer the questions posed in the previous section. A complete overview of all features considered in our experiments is given in Table 1, including the maximum, mean and variance across all active students in all periods. Not all features are used for each experiment, see section 3.

All features are aggregates over the activity period. Below follows a detailed description:

- f_1 describes the activity during the period of day, where Danish students are normally in school, while f_2 describes the activity during non-school hours.
- f_3, f_4 and f_5 describe time spent doing exercises, reading texts and taking quizzes respectively.
- f_6, f_7 and f_8 describe time spent working with different topics: languages (Danish, English, German), societal (social studies, history, etc.) and science (physics, biology, etc.), respectively.
- f_9 is the average session length during the activity period.
- f_{10} is the average quiz score; this feature may be missing, if a student takes no quizzes during an activity period, but our analysis methods can handle this, see section 2.2.
- f_{11}, f_{12}, f_{13} and f_{14} describe the time spent doing exercises of different complexity, measured by their level in Bloom's taxonomy. We regroup the levels of Bloom's taxonomy into 4 levels:

  f_{11} **Remember/Understand**: Exercises involving reading and describing, e.g. "Read a map".

  f_{12} **Apply**: Exercises involving application of previously learned concepts, e.g. "Practice adjectives".

  f_{13} **Analyze/Evaluate**: Exercises involving discussion, analysis and experimenting, e.g. "Work with the poem", "Analyze the game".

  f_{14} **Create**: Exercises involving creation of a product, e.g. "Create a cartoon", "Write a story".

Having extracted m features for each student in each period, we construct the matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$, where each of the n rows consists of the feature vector for an active student in a given activity period. Thus each student occurs several times in $\mathbf{X}$; once for each period, where they are active.

2.2 Soft Clustering using Non-negative Matrix Factorization

We will utilize non-negative matrix factorization for our soft clustering. The use of NMF as a soft clustering technique has become popular in recent times [10], with applications within several fields, such as clustering of images and documents [8, 13]. NMF has also seen success in the educational data mining community, for clustering tasks, as well as other tasks such as performance prediction [3, 12].

i	f_i	Max	Mean	Variance
1	Hours between 8AM and 4PM	31.85	0.940	0.862
2	Hours before 8AM and after 4PM	71.84	0.174	0.283
3	Hours doing exercises	3.61	0.048	0.019
4	Hours reading texts	7.73	0.344	0.148
5	Hours taking quizzes	23.76	0.231	0.297
6	Hours working with language subjects	58.28	0.531	0.693
7	Hours working with societal subjects	45.96	0.294	0.285
8	Hours working with science subjects	103.69	0.277	0.326
9	Average session length in hours	7.91	0.268	0.027
10	Average quiz score (in [0, 1])	1.00	0.733	0.034
11	Hours working with Bloom level 1	2.83	0.016	0.006
12	Hours working with Bloom level 2	1.64	0.008	0.002
13	Hours working with Bloom level 3	1.51	0.014	0.003
14	Hours working with Bloom level 4	2.04	0.009	0.003

Table 1: Overview of features.

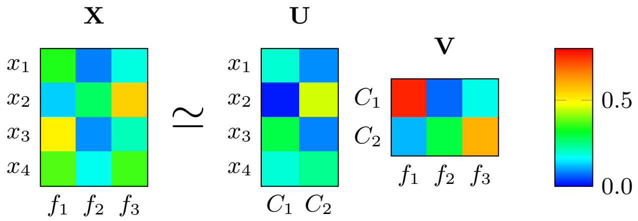


Figure 2: The soft clustering given by NMF.

NMF is a dimensionality reduction method, in which we are given a non-negative matrix $\mathbf{X} \in \mathbb{R}_+^{n \times m}$ and $k \in \mathbb{N}$, and wish to determine $\mathbf{U} \in \mathbb{R}_+^{n \times k}, \mathbf{V} \in \mathbb{R}_+^{k \times m}$, such that $\mathbf{X} \simeq \mathbf{UV}$. More specifically, we search for $\mathbf{U}$ and $\mathbf{V}$, such that the error $||\mathbf{X} - \mathbf{UV}||_F$ is minimized, where $|| \cdot ||_F$ is the Frobenius norm. For our analysis, we need to be able to handle missing values in $\mathbf{X}$. In this case the NMF problem is reformulated as the *weighted non-negative matrix factorization*, in which we are also given a binary weight matrix $\mathbf{W} \in \{0,1\}^{n \times m}$, where a 0 indicates missing data. Now, we wish to find $\mathbf{U}, \mathbf{V}$ such that $||\mathbf{W} \odot (\mathbf{X} - \mathbf{UV})||_F$ is minimized[2].

$\mathbf{U}$ and $\mathbf{V}$ admits a soft k-clustering as shown in Figure 2; $\mathbf{V}$ describes the importance of each feature for each cluster (for instance, f_1 has high importance in C_1), while $\mathbf{U}$ describes the membership of each data point to the different clusters (for instance, x_3 is mostly in C_1, while x_4 is in both clusters).

Note, that for NMF, we have $\mathbf{X} \simeq \mathbf{UV} = \mathbf{UIV} = \mathbf{UA}^{-1}\mathbf{AV}$, where $\mathbf{I}$ is the $k \times k$ identity matrix and $\mathbf{A}$ is a $k \times k$ invertible matrix. This means that we may rescale $\mathbf{U}$ and $\mathbf{V}$ by this matrix, $\mathbf{A}$, and its inverse. In our analysis, we use this to rescale $\mathbf{V}$, such that all rows of $\mathbf{V}$ (the clusters) sum to one, thus making the clusters comparable, and membership of the different clusters easier interpretable.

There exist several algorithms for obtaining the non-negative matrix factorization of $\mathbf{X}$, for instance basic gradient descent, multiplicative update rules and alternating least squares; [1] gives a good overview in the non-weighted setting. Several of these algorithms have been adapted for the WNMF case, while approaches based on *expectation maximization* have also been proposed, see [6]. For our analysis, we will use the weighted version of the multiplicative update method, proposed by Lee and Seung [9].

The NMF algorithm given in [9], adopted to WNMF [6], is as follows:

1. Initialize $\mathbf{U}$ and $\mathbf{V}$.
2. Repeatedly update $\mathbf{U}$ and $\mathbf{V}$ by the following rules:

$$\mathbf{U} \leftarrow \mathbf{U} \odot \frac{(\mathbf{W} \odot \mathbf{X})\, \mathbf{V}^T}{(\mathbf{W} \odot (\mathbf{UV}))\, \mathbf{V}^T}$$

$$\mathbf{V} \leftarrow \mathbf{V} \odot \frac{\mathbf{U}^T\, (\mathbf{W} \odot \mathbf{X})}{\mathbf{U}^T\, (\mathbf{W} \odot (\mathbf{UV}))}$$

where division is done element-wise.

The literature explores several ways of initializing $\mathbf{U}$ and $\mathbf{V}$; in our case, we will simply use random initialization. The alternating optimization steps are applied until the decrease in error reaches below a set threshold. Finally, Lin has noted that the procedure described above may not converge to a stationary point, hence we modify the update rules as proposed by them [11]. Furthermore, since we in our case know all missing values of $\mathbf{X}$ to be bounded by a constant c, we modify the above procedure such that 0-weight values of $\mathbf{UV}$ that deviate above c are penalized, i.e. whenever a value $(\mathbf{UV})_{ij}$ with $\mathbf{W}_{ij} = 0$ gets larger than c, we set $\mathbf{X}_{ij} = c$ and $\mathbf{W}_{ij} = 1$, before the next update step. If $(\mathbf{UV})_{ij}$ decreases below c again, the weight is reset to 0.

It remains to be seen, how we select the number of clusters, k. For each experiment, we construct clusterings with $k = 1, 2, ...,$ and stop when the decrease in error going from k clusters to $k + 1$ clusters is below some threshold, which depends on the initial error. As a consequence clusters will be uncorrelated on a student level, since otherwise we would pick a lower k.

[2] $\odot$ denotes the Hadamard product (element-wise multiplication).

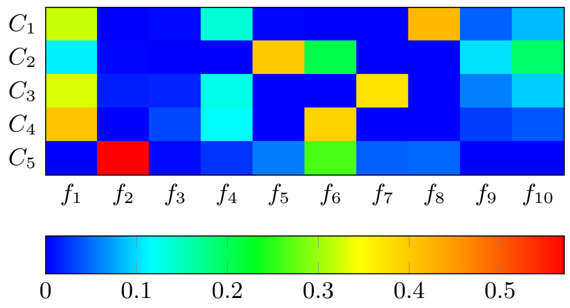


Figure 3: The cluster matrix for the first experiment.

3. EXPERIMENTS AND RESULTS

In this section, we present two different experiments using the setup described above. In the first experiment, we investigate the relation between activity, activity type, subject, time of day, average session length and performance. In the second experiment, we investigate the relation between complexities of exercises and subjects.

3.1 Performance and Optimal Behavior

In the first experiment, we investigate the relation between activity, activity type, subject, time of day, average session length and performance, i.e. we consider features $f_1, ..., f_{10}$. The features are extracted and $k = 5$ is selected, as described in section 2. We run the WNMF algorithm, and obtain the cluster matrix V as shown in Figure 3. From the figure, we can make several observations about the clusters:

C_1 In this cluster, we find students mostly working with the science subjects (f_8). These students seem to work mostly during school hours (f_1). The students also seem to spent a lot of time reading (f_4).

C_2 Students in this cluster spend a lot of time taking quizzes (f_5). They will spend some time during school hours (f_1) and some time working with language subjects (f_6). Furthermore, students in this cluster seem to both have fairly long average session length and high performance (f_9 and f_{10}).

C_3 In cluster C_3, we see students working with societal subjects (f_7). They work during school hours (f_1) and spend time reading texts in the system (f_4).

C_4 This cluster shows a relationship between being active in school (f_1) and spending time in the language subjects (f_6). Students in this cluster also spend time reading texts (f_4) and doing some exercises (f_3).

C_5 The most important feature for C_5 is f_2, i.e. the students in this cluster spend most time using the system during non-school hours. These students spent time in all subjects, but mostly languages (f_6), and they spent time taking quizzes (f_5).

From the clusters, we can see that the impact on performance from different behaviors depends on the subject. From cluster C_2, we see that students working mostly with language subjects gain most performance from spending time taking quizzes and working during school hours, whereas students working mostly with societal (cluster C_3) and science (cluster C_1) subjects gain most from reading texts, while working mostly during school hours. Note that cluster C_4 indicates that students working with languages may also improve performance by reading texts, but to a lesser degree than students working in other subjects. Finally, C_5 indicates that working mostly from home and primarily taking quizzes, does not improve performance. While C_5 indicates this for all subjects, the high importance of f_4 indicates that this most often occur for students working with languages, confirming the observations from C_2. Finally, it is also worth noticing, that there is a strong relation between performance and average session length (clusters C_1, C_2 and C_3), indicating that students, who perform well, also have longer sessions on average.

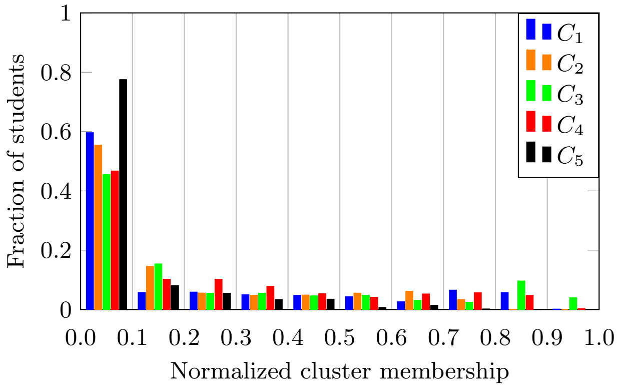


Figure 4: The distribution of cluster membership for the first experiment.

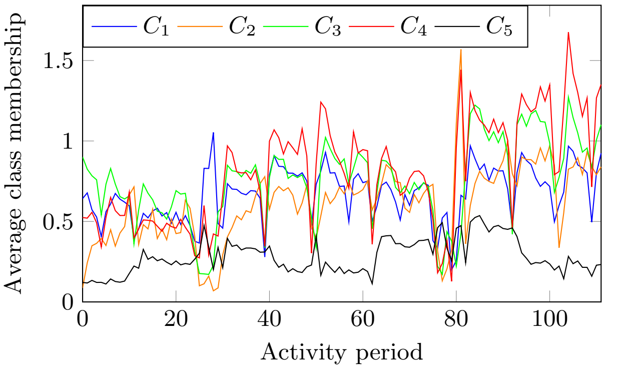


Figure 5: The average cluster membership in each activity period for the first experiment.

From the above discussion, it appears that the behavior in clusters C_4 and C_5 are sub-optimal, when considering performance, while students gain more from being in C_1, C_2 or C_3, i.e. by working during school hours, having longer sessions and taking quizzes (in the case of languages) or reading texts (in the case of societal or science subjects).

Figure 4 describes the distribution of cluster membership across all students and all activity periods , i.e. the columns of the first interval $[0, 0.1)$ gives for each cluster the fraction of students with 0%-10% membership. We see, that we do indeed get a soft clustering, with students often belonging to more than one cluster. Only C_3 seems to be the sin-

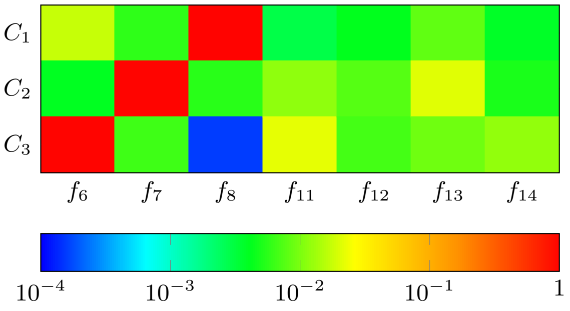


Figure 6: The cluster matrix for the second experiment. Note, that a logarithmic scale is used for this plot.

gle dominant cluster of some students. From the figure, we also see that students are typically never exclusively in C_5, which is positive, as the behavior observed in that cluster was not very productive in terms of performance. Other than that, we generally observe that students seem to distribute fairly well between the top four clusters, indicating most time spent during school hours and a varied use of both quizzes and texts across all subjects.

Next, we analyze how the membership of different clusters change over time. Figure 5 plots the average membership for each period, i.e. the average of rows from **U** belonging to the given period. The first observation we make from Figure 5, is that clusters C_1, C_2, C_3 and C_4 appear correlated at the system-wide level. This is due to these clusters being dependent on the general activity in the online system; most of the sudden drops occur at the same time as Danish school vacations, most notably the two larger drops around activity periods 25 and 79 (see Figure 1). C_5 seems to be relatively unaffected by the general activity, but this makes sense, as C_5 contains mostly students, who work outside school hours, and thus a lower membership is expected in that cluster in general, which is also the pattern we see in periods with no vacation.

Looking at the general distribution between the different clusters, C_3 and C_4 seem to be the most dominant, indicating that most students are working with language and societal subjects and reading texts. Cluster C_1 (science subjects) is fairly constant in the non-vacation periods, and C_2 seems to increase starting period 80, indicating that more students spend time taking quizzes. Finally, as mentioned, C_5 is the least active cluster across most periods. One general trend for the top four clusters seem to be an increase in activity during the 112 periods, indicating that students are spending more time in the system on average.

3.2 Subject and Exercise Complexity

In the second experiment we look at the relation between subjects and exercises grouped by Bloom's taxonomy level, i.e. we consider features $f_6, f_7, f_8, f_{11}, f_{12}, f_{13}, f_{14}$

We expect three clusters, one for each of the subject classes, which will tell us how much each Bloom level is used within each subject class. Figure 6 shows the cluster matrix found. From Figure 6, we make the following observations:

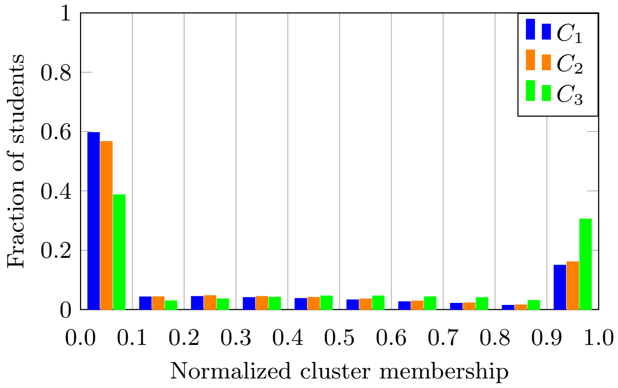


Figure 7: The distribution of cluster membership for the second experiment.

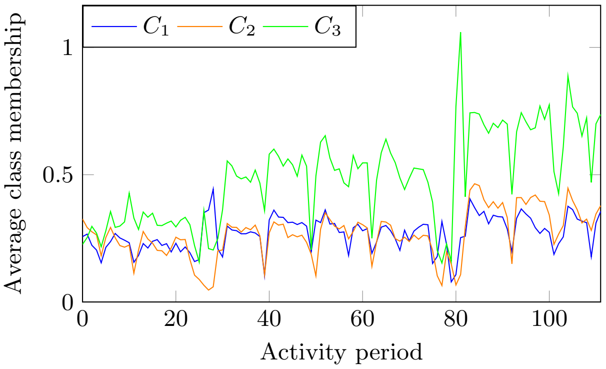


Figure 8: The average cluster membership in each activity period for the second experiment.

C_1 In the science subjects, only very little of the 3 higher levels are used, and almost none of reading and understanding.

C_2 For societal subjects, students have only little activity in the first 2 levels, a lot in analyzing and evaluating, and very little activity in creation.

C_3 In languages, students have a tendency to read and understand a lot, and then distribute almost evenly on the 3 higher levels.

This implies that if we want to attract students to use an online educational system for languages, focus should be on exercises with Bloom's taxonomy level read and understand. For societal subjects the focus should be on exercises with analyzing and evaluating. For science we see no preference.

From Figure 7, we see that the clustering has many high values which is most likely explained by having a teacher who uses the system exclusively in only one of the subjects, which we can see happens most often for languages.

As we can see in Figure 8 all three clusters share similar curvature, which is partly explained by holidays. Especially the science and societal clusters behave seem highly correlated on a general level. We also see that in all three subjects, the average time spent during a week has gone from 15 minutes, to 45 minutes for languages and 25 minutes for both societal subjects and sciences. A clear indication that teachers and students in Denmark are using online educational systems more, especially for languages.

4. CONCLUSIONS AND FUTURE WORK

Several points can be taken from our analysis. We have identified three optimal and two sub-optimal behaviors in relation to subject and performance. One notably conclusion is that students using the Clio Online system during non-school hours (at home) do not seem to gain any significant boost to performance. We also saw how taking quizzes seems to increase the performance of students in languages, more so than in other subjects, where reading texts are of more importance. This fits the intuition that skills such as grammar need to be trained, in order to be learned. We inform how exercises are used depending both on their subject and their level in Bloom's taxonomy. And lastly we see that the average amount of time spent in the system is increasing both generally and for the individual students in all subjects, but especially for students working with languages. Furthermore, both experiments show how behaviors can have high correlation on a system-wide level, despite being uncorrelated on the individual student level. While the change of behavior for individual students was not directly analyzed in this paper (due to privacy concerns), our method allows for tracking such individual changes, hopefully helping teachers encourage optimal student behavior, e.g. by recommending training quizzes for students working with languages, or making sure that students are allowed more time to use the system in school.

In our setting, the number of clusters is fixed. It may be interesting to use an adaptive clustering strategy instead, as done in [7], as one might expect clusters to change over time. In the future, it might also be interesting to include other features, that were not available to us at this time, for instance whether a text (or quiz) have been assigned by a teacher, or whether the student reads it by themselves. For this study, we also only had access to a limited amount of data; better and more reliable results might be obtained by including more data.

5. ACKNOWLEDGMENTS

The work is supported by the Innovation Fund Denmark through the Danish Center for Big Data Analytics Driven Innovation (DABAI) project. The authors would like to thank Clio Online, and the reviewers for their thorough and insightful feedback.

6. REFERENCES

[1] Michael W. Berry, Murray Browne, Amy N. Langville, V. Paul Pauca, and Robert J. Plemmons. Algorithms and Applications for Approximate Nonnegative Matrix Factorization. *Computational Statistics & Data Analysis*, 52(1):155 – 173, 2007.

[2] Louis Faucon, Lukasz Kidzinski, and Pierre Dillenbourg. Semi-Markov model for simulating MOOC students. In *Proceedings of the 9th International Conference on Educational Data Mining (EDM)*, pages 358–363. International Educational Data Mining Society (IEDMS), 2016.

[3] Ben U. Gelman, Matt Revelle, Carlotta Domeniconi, Kalyan Veeramachaneni, and Aditya Johri. Acting the Same Differently: A Cross-Course Comparison of User Behavior in MOOCs. In *Proceedings of the 9th International Conference on Educational Data Mining (EDM)*, pages 376–381. International Educational Data Mining Society (IEDMS), 2016.

[4] Christian Hansen, Casper Hansen, Niklas Hjuler, Stephen Alstrup, and Christina Lioma. Sequence modelling for analysing student interaction with educational systems. In *Proceedings of the 10th International Conference on Educational Data Mining (EDM)*, pages 232–237. International Educational Data Mining Society (IEDMS), 2017.

[5] Stephen Hutt, Caitlin Mills, Shelby White, Patrick J. Donnelly, and Sidney K. D'Mello. The Eyes Have It: Gaze-based Detection of Mind Wandering during Learning with an Intelligent Tutoring System. In *Proceedings of the 9th International Conference on Educational Data Mining (EDM)*, pages 86–93. International Educational Data Mining Society (IEDMS), 2016.

[6] Yong-Deok Kim and Seungjin Choi. Weighted Nonnegative Matrix Factorization. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1541–1544, 2009.

[7] Severin Klingler, Tanja Käser, Barbara Solenthaler, and Markus Gross. Temporally Coherent Clustering of Student Data. In *Proceedings of the 9th International Conference on Educational Data Mining (EDM)*, pages 102–109. International Educational Data Mining Society (IEDMS), 2016.

[8] Cosmin Lazar and Andrei Doncescu. Non Negative Matrix Factorization Clustering Capabilities; Application on Multivariate Image Segmentation. In *Proceedings of the 3rd International Conference on Complex, Intelligent and Software Intensive Systems (CISIS)*, pages 924–929, 2009.

[9] Daniel D. Lee and H. Sebastian Seung. Algorithms for Non-negative Matrix Factorization. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pages 556–562, 2000.

[10] Tao Li and Chris Ding. Non-negative matrix factorization for clustering: A survey. In *Data Clustering: Algorithms and Applications*, pages 149–176. Chapman & Hall/CRC, January 2013.

[11] Chih-Jen Lin. On the Convergence of Multiplicative Update Algorithms for Non-negative Matrix Factorization. *Trans. Neur. Netw.*, 18(6):1589–1596, 2007.

[12] Stephan Lorenzen, Ninh Pham, and Stephen Alstrup. On Predicting Student Performance Using Low-rank Matrix Factorization Techniques. In *Proceedings of the 16th European Conference on e-Learning (ECEL)*, pages 326–334. Academic Conferences and Publishing International, 2017.

[13] Farial Shahnaz, Michael W. Berry, Victor P. Pauca, and Robert J. Plemmons. Document clustering using nonnegative matrix factorization. *Information Processing & Management*, 42(2):373 – 386, 2006.

The influence of task activity and the learner's personal characteristics on self-confidence during an online explanation activity with a conversational agent

Yugo Hayashi
Ritsumeikan University
2-150 Iwakura-cho,Ibaraki
Osaka,567-8570,Japan
y-hayashi@acm.org

Yugo Takeuchi
Shizuoka University
836,Ohya,Suruga-ku,Shizuoka
Shizuoka,422-8529,Japan
takeuchi@inf.shizuoka.ac.jp

ABSTRACT

This study investigated the factors underlying the estimation of learner self-confidence during explanations with a conversational agent in an online explanation task. Based on reviews of previous studies, we focused on how factors such as the learner's task activities and personal characteristics can be predictors. To examine these points, we used an online explanation task, which was run by a conversational agent for students in a classroom on information processing psychology (n=99). We asked the participants to make text-based explanations to the agent in a question-and-answer (Q&A) style, and clarified a particular concept that was taught in a previous lecture in the class. The results show that an increase in the amount of actual task work for explanations and personal characteristics (such as social skills) helped to predict higher self-confidence. The findings have implications not only for knowledge of how such factors might influence learner self-confidence in an online explanation task, but also for the design of online tutoring systems that can automatically detect learner confidence using these variables, and facilitate learning adequately based on such data.

1. INTRODUCTION

Networked learning such as the use of massive open online courses (MOOCS) and tutoring systems, which include social networking services (SNS) has seen many advances in recent years and has become a popular way of supporting learning through social interaction. Such environments allow learners to interact with each other through conversation, and have drawn the attention of many socio-constructionists in the field of learning science. Numerous investigations in this field focus on discussion boards [5, 25], and an emerging number of studies have examined the technological side of research. Moreover, these studies have explored how to detect the learner's conversational behavior. Researchers in artificial intelligence education (AIED) have been investigating the use of conversational agents (CAs) in online environments [20] and have explored the use of agents that play the role of peer learner, whereby they interact socially as discussants in a serious game-based environment [20]. Some research on online tutoring systems examines the use of agents that play the role of the student, whereby learners absorb information through teaching the agents [16, 17]. One of the most important points of learning by teaching is that the learner can reflect on his ideas by observing his externalized thoughts. In the context of social learning, metacognitive abilities might help him identify the perspectives of other learners/agents to establish shared knowledge and successfully coordinate with one another.

Despite concerns surrounding the effects of social learning on social coordination skills and metacognitive abilities, not many experimental investigations have explored the learner's task efficiency and metacognitive process, such as confidence during interactions with a CA. Our study centers on the learner's metacognitive capacity; for example, in relation to confidence evaluations in an explanation activity with a CA. We investigated how the learner's task activity and personal characteristics impact his confidence level during tasks, and propose a model to understand learner confidence during online tutoring with an agent. We also discuss how our model could predict learner confidence, and subsequently develop an automated tutoring system that can collaboratively respond based on learner confidence.

1.1 Conversational Agents and their use in online Learning

The number of studies on computer-based learning that employs intelligent tutoring systems has grown rapidly over the past three decades[14, 27]. Advances in language technology have enabled the development and use of CAs, which can act as peers learners or mentors, and have made progress in terms of facilitating learning activities[10, 8]. Initial studies focused on the use of embodied CAs that act as educational companions or tutors and facilitate the learning process as it relates to motivation[4]. Moreover, recent research has examined the implications of such technology on learning gains through learning by doing [1]. Many studies investigate the use of agents capable of handling natural conversation; these agents are developed based on conversational dialog models, and have demonstrated the successful use of tutoring in social interactions. One example is AutoTutor, a system that allows students to engage in conversations for their projects. Recently, more advanced online tutoring systems, such as Operation ARIES!, have employed CAs [20] where learners absorb information through web-based tasks in which they talk with CAs. Other tutoring systems have begun to incorporate elements such as SNS [13]. In such cases, learners can interact with other learners and CAs; they must use metacognition to monitor their own perspectives as well as those of their peers in order to better coordinate with one another. Many important psychological issues have not yet been explored in depth; for example, how learners develop their confidence by reflecting on activities in such an environment. In the next section, we will look at some of these points based on reviews of related studies.

1.2 Self-confidence and learning

The 2015 report of the Programme for International Student Assessment (PISA) an initiative of the Organisation for Economic Co-operation and Development (OECD) identifies several types of skills such as prior knowledge, personal characteristics, collaborative capacity, and problem-solving skills. These abilities were assessed using pedagogical CAs, which acted as peer learners and tutors. The report mentions self-monitoring as an important skill because learners must be able to keep track of how their abilities, knowledge and perspectives affect their interactions with other agents in relation to the task at hand [23]. Monitoring skills can be detected through evaluations of self-confidence; this issue has been broadly examined in cognitive psychology.

Cognitive psychology research has a long history of studying metacognition, such as self-monitoring of task efficiency, which is deeply linked to performance [19]. According to the literature, problem-solving involves conscious, step-by-step observation of one's problem-solving behavior. Throughout this process, one can estimate the likelihood of the ongoing task having success or failure. In terms of learning activities, high confidence is known to reflect higher quality mental representations of a task, and is associated with long-term recall [7]. In this sense, a hypothesis can be deduced, such as that the actual task activity might facilitate the learner's monitoring; for example, regarding the self-evaluation of one's confidence about a task. Interestingly, some educational psychology studies have revealed that self-assessments of learning achievement are negatively correlated with learning performance [9]. One explanation for this outcome might be that learners have inherent cognitive limits that hinder simultaneous monitoring and execution of a task. They might also have individual differences in terms of their capacity to self-monitor. Additional types of individual skills that can be captured by self-assessments might play a role in self-monitoring. In this context, we investigated participants' ability to self-monitor their confidence about a task activity. Next, we analyzed the relationship between self-monitoring and the personal characteristics, which might also affect confidence level.

1.3 Personal characteristics and Learning

Along with concerns raised in the previous section about personal characteristics, recent reports have shown that qualities such as attitude, interpersonal skills, personal traits, and motivation can influence individual learning activities[23]. Studies examining such personal features have shown that these factors indeed influence learning; for example, when it comes to thinking style [26]. In the context of this study, where learners interact socially with an agent, it is important to focus on the learner's personal qualities as they relate to social interaction and communication skills. Some research has explored the use of Big-Five questionnaires [15], which center on personal characteristics, such as social skills. Previous studies have indicated that learners with poor social skills might have lower collaborative performance [21]. Other studies by [12] have investigated how learners' skills influence their performance during an online, concept-learning tutoring task with a pedagogical Conversational Agent (PCA). During in this task by [12], learners were guided by a PCA that helped them formulate their explanations of a key concept taught in a large-scale class. The results show that learners with higher social skills performed better on explanation activities with the PCA. Taking this into consideration, personal characteristics such as social skills will also influence metacognitive states, which are related to task performance. Thus far, no investigations have delved into the relationship between social skills and self-confidence; however, this study does. Based on this, we focus on a particular situation whereby most studies using agents have not yet fully examined the influence of personal characteristics on learning activities.

1.4 Goal and Hypothesis

This study investigates how the learner's task work influences metacognition of his/her work, and consequently, self-confidence. Furthermore, we examined how personal characteristics, which are considered important for inter-personal interactions, impact both the task activity and the learner's metacognition of the task. To explore these points, we used an online explanation task where we asked learners to give explanations to a social CA in a Q&A style, and to chat about a particular concept that was taught in a previous lecture. Based on reviews of previous research on learning activities and metacognition, we hypothesized that an increase in the amount of actual task work, such as giving many explanations to an agent, would enhance self-confidence about one's work (H1). For our second goal, we focused on the relationships between personal characteristics and work on explanation activities, as well as the learner's metacognition of that work. We posited that higher interpersonal skills would increase the number of actual explanation activities in relation to the social agent (H2-a), and would also enable metacognition of the student's explanations (H2-b). In the next section, we will demonstrate how we analyzed these points.

2. METHOD

2.1 Participants and conditions

Ninety-nine (Mage: 20.52, SD: 1.60) Japanese university students majoring in psychology participated in this study. The students, whom we call learners, were taking a lecture class on information processing psychology in 2014 and used the system as part of their coursework. The learners were taught about 30 basic concepts of human information processing such as top-down processing, neural networks, Bayesian models, and expert systems.

2.2 Procedure

After the participants attended lectures about the basic concepts taught in class, they took part in an online tutoring task that was valid for two weeks. They logged into the web system using their ID and password, and worked on the task based on their personalized page. They could only access the system on campus using the computer terminals located there. Only members of the class were registered in the system and were assigned to groups consisting of 4-5 students. In each group, the participants worked on the same materials, and the system provided them with updated information about their fellow members.

The aim of the task was to facilitate learner's self-explanations[6] of the basic ideas they learned in class by conversing with social agents though online texts. As they began the task, the agent appeared on their screens and asked them questions about a specific concept. The questions consisted of 17 types such as: "Can you explain the key term regarding how it functions? " " How do you use it in your daily life? " and " Can you think of a concept similar to this one? " Learners were able to restart and continue the task, even if they terminated it during the Q&A session. After they answered one question, the page switched to an assessment page, and the system asked them to assess their confidence level. As will be explained in the following section, this was done to measure the degree of self-confidence. Afterward, learners received feedback from the social agent, along with examples from other members when inputs were entered into the database. If there was no updated information from classmates, the system used a database from the previous year instead. As learners finished answering all 17 questions, they completed the task. This activity lasted an average of 30 minutes.

2.3 System

The system was operated on an Apache web server via a CentOS server. The scripts of the web pages were written in PHP, JavaScript, HTML and CSS. MySQL was used for the database. This system is a modified version of [11] and is called " Web-based Explanation Support with Conversational Agent " (WESCA). The system has a database that manages thirty different key terms that were selected from the class material; one was assigned to each of the learners according to their ID numbers. The agent in the system responds to the learner's text sentences and the number of questions based on the bag-of-word method. The system can also retrieve other members' answers (using them as examples) from the logs based on year, and data from previous years if there is no updated information. The system also features social awareness functions such as evaluating the other learners pushing the " like " button. The current version does not have any functions to show learners how many likes they have received during the task.

2.4 Measures

This study focuses on three factors: (1) degree of self-confidence while interacting with the agent, (2) the amount of interaction with the agent, and (3) the effects of personal characteristics on social skills. In the following section, we describe the types of measures that we used to capture these factors.

2.4.1 Meta cognition: Self-confidence

To capture learner self-confidence during the participants' explanations, we collected assessments based on confidence level about the explanations for each Q&A session with the agent. Learners were required to input their self-confidence level based on a seven-point scale (-3: not very confident to 3: very confident). As with the number of interactions, we analyzed the average level of confidence for each individual, and used these levels as representative values for each participant.

2.4.2 Number of interactions: The amount of words used to explain

We calculated the number of interactions based on the number of words that the learners input while responding to the agent. For each individual learner, we used the average number of words that were input into the system as a representative value for the number of interactions with the agent.

2.4.3 Personal characteristics: The autism spectrum quotient (AQ) score

We assessed the degree of social communication skills based on the questionnaire, which was originally developed in [3] and translated into Japanese. This questionnaire appraises social skills based on the autism spectrum quotient (AQ) and was originally was used to investigate whether healthy adults had symptoms of autism. The questionnaire consists of 50 questions covering five different domains associated with the autism spectrum: (1) social skills, (2) attention switching/tolerance for change, (3) attention to detail, (4) communication skills, and (5) imagination. For each question, learners assessed how strongly they felt about themselves on a five-point scale (1: Doesn't match, 5: Does match). For example, a question about social skills would be, " I like to do activities that require interacting with others.' " The higher the score, the lower the learner's degree of autism, which indicates strong social communication skills. For each learner, we calculated the five factor scores of domains (1) to (5) using factor analysis, and used this as the representative value for analysis.

3. RESULTS

To examine our two hypotheses, we first explored how learners' explanations that they gave to the agent influenced their self-efficiency. For this point, we investigated the relationships between (1) the number of explanations given to the agent and the degree of learner self-confidence regarding the achievement of the activity. Then, we looked at how individual characteristics (such as social communication skills) influenced both the number of explanation activities and self-confidence. For this aspect, we analyzed (2) the relationship between the AQ scores and the number of explanations and confidence levels.

3.1 Explanation activities and self-confidence

First, we conducted a correlation analysis using the Pearson correlation coefficient to identify any relationships between the two variables, as well as the average number of words used during the explanation activity, and the average confidence level about the explanation given to the agent. The findings show that there were significant relationships between the two variables ($r = 0.211$. $p < .05$). Figure 1 describes the correlations between the two variables. Next,

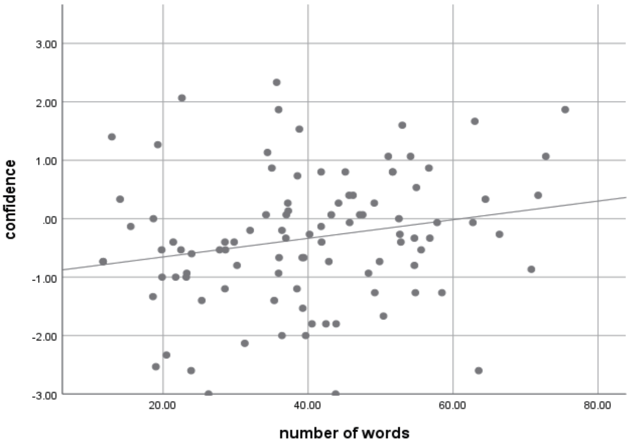


Figure 1: Relationship between learner's confidence and typed words.

Table 1: Results of correlations between personal characteristics and number of words

	# of words
1. social skills	0.051
2. attention switching	-0.023
3. attention to detail	0.149
4. communication skills	0.076
5. creativity	0.034

we explored how explanation activities influenced confidence level by conducting a single regression analysis. We employed the evaluations of confidence as the dependent variable, and number of words used during explanations as the independent variable. We used the forced entry method to perform the analysis and acquired the regression equation with the coefficient of determination (R^2=0.035 by $p < .05$). These results suggest that the actual performance of interactions (such as explanation activities) facilitates metacognition, thus supporting hypothesis H1.

3.2 Personal Characteristics

3.2.1 Personal Characteristics and explanation activities

Next, we analyzed the correlations between the scores of the five domains and the types of words to see how the personal characteristics considered by the AQ questionnaires related to task activity. More specifically, we examined the correlation between the number of words used for the explanations and each of the five AQ domain factors: (1) social skills, (2) attention switching/tolerance for change, (3) attention to detail, (4) communication skills, and (5) imagination. Table 1 depicts the correlations between the variables.

The outcomes of the analysis of the Pearson correlation coefficient revealed no significant links between any of the AQ categories. This indicates that personal qualities captured from the AQ scores do not have any influence on explanation activities with the agent. This shows that hypothesis H2-a was not supported.

Table 2: Results of correlations between individual characteristics and learner's confidence

	leanrer's confidence
1. social skills	0.312
2. attention switching	0.211
3. attention to detail	-0.164
4. communication skills	0.170
5. creativity	-0.025

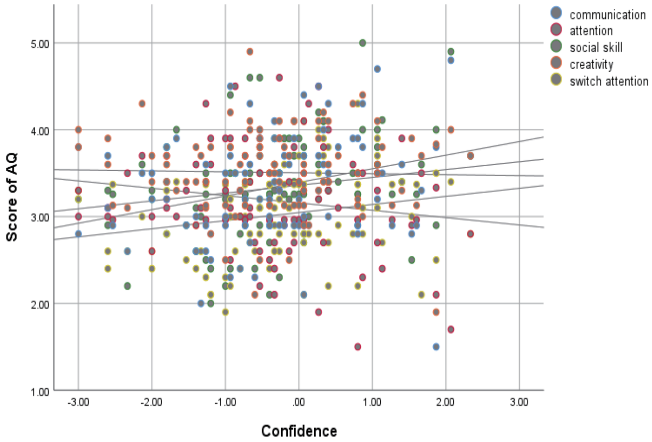


Figure 2: Relationship between each AQ score (Y-axis) and learner's confidence(X-axis).

3.2.2 Personal Characteristics and self-confidence

Next, to see how personal characteristics considered by the AQ scores were related to confidence, we looked at the correlations between the AQ scores and confidence level. For each of the five AQ factor scores, we explored the correlation with confidence level. Table 2 shows the correlations between the variables. The findings of the analysis of the Pearson correlation coefficient revealed significant links between confidence and three variables from AQ: (1) social skills (r = 0.312. $p < .01$), (2) attention switching/tolerance for change (r = 0.211. $p < .05$), and (4) communication skills (r = 0.170. $p < .05$). To see if personal characteristics considered by the AQ scores influenced the number of explanation activities, we conducted a multiple regression analysis. Figure 2 shows the outcomes of the correlations between the two variables. We employed the number of words as the dependent variable and the five AQ factors as the independent variables. We used the forced entry method to perform the analysis, and acquired a regression equation with the coefficient of determination R^2=0.137 by $p < .05$. Table 3 shows the summary of the multiple regression analysis. These findings demon-

Table 3: Summary of the mutiple regression analysis

	regression coefficient B
1. social skills	.368
2. attention switching	.044
3. attention to detail	-.041
4. communication skills	.004
5. creativity	-.203

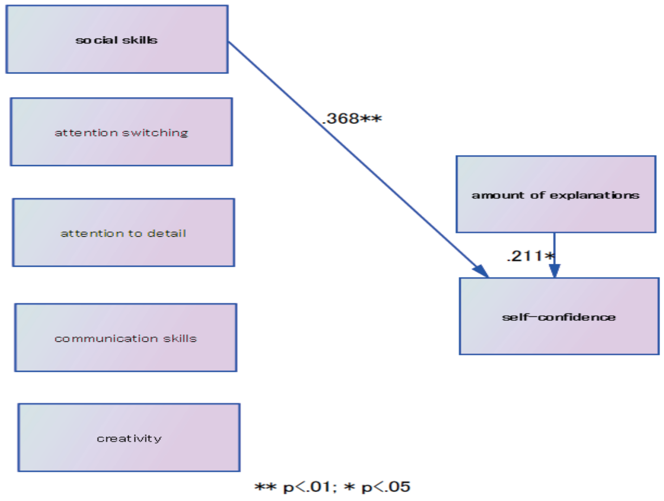


Figure 3: Overall results of AQ score amount of explanations and self-confidence. Indices's indicate the regression coefficient B.

strate that only the variable of social skills influenced learner evaluations of confidence. This indicates that personal characteristics influence learners' metacognition of their learning activities, thus supporting hypothesis H2-b. Figure 3 portrays the summary of the results, including path variables. This figure shows the model of how personal characteristics and actual task work facilitate self-confidence. Our data analysis suggests that this model could potentially predict learner self-confidence, which we will discuss further in the next section.

4. DISCUSSION

4.1 Developing an automatic tutor to detect learner self-confidence

Our results show that self-confidence was related to the task activity, as well as personal traits (such as social skills). Considering these findings, the actual amount of words input and previously self-evaluated personal scores can help predict confidence level. Therefore, we can use this model to develop systems that can become aware of the learner's subjective states. We can also employ it to design pedagogical agents that could prompt learners to request help or encourage those with low confidence. As discussed earlier, learner self-confidence is highly related to task performance[7]; identifying learners' cognitive states might facilitate self-efficiency [2] during the task, which could result in higher learning performance. Discussions about learning performance could go beyond the topic of this paper. I would like to show how the proposed model could be used to automatically predict learner self-confidence during the task. For this investigation, we used machine learning to see how the categorical factors that were extracted from the previous analysis might be optimal for detection. We used linear discriminant analysis (LDA), using confidence as the dependent variable and the number of words and social skills score as independent variables. Confidence was labeled as a binary of high/low based on the median of the acquired data set. The results of the LDA show an accuracy rate of 66.7%, which indicates a relatively high validation of categorization. There have been recent attempts to detect and model self-efficiency in tutoring systems[18]. However, not many studies focus on personal characteristics as predictors. In this sense, our model could provide a new way to capture learners' subjective states. However, as noted above, more integrated investigations should be carried out, along with an analysis of learners' performance during the explanation activities. To do so, we should evaluate learners' output messages and see how they relate to the variables acquired in this study.

4.2 Motivating learners via socialized feedback from the conversational agent

The system used in this study features functions such as providing feedback about other group members' explanations. Moreover, learners were able to assess each other's explanations by clicking on the “like” buttons, as in SNS. These social functions are adequate for motivating learners and reducing the dropout rate. One of the methods used to facilitate learner self-efficiency in such educational environments could be designed by providing feedback, such as how many “likes” they receive during their activities. Telling learners that they have been nominated as good explainers in the group is another way to motivate them. The CA can provide such feedback, as it is well-known that people can praise each other in human-computer interactions [22]. Related studies from our research group have been developing systems through which students can request help online, as well as systems that support teachers in programming classes[24]. Learners in the classroom use the system and report the ongoing progress of their programming tasks. As they complete each task, an agent installed in the system contacts the learner and sends a request for him to help other classmates who are still stuck working on a problem. The system aims to increase learners' self-esteem by approving/selecting him to help his classmates. The study focuses on motivation when a learner becomes a teacher, as well as on learning in the domain of programming skills. In future, the system to be introduced in this current study might utilize such features, the goal being to encourage learners to use these types of help-requesting functions provided by CAs.

5. CONCLUSIONS

This study focused on self-confidence during explanations with a CA in an online explanation task. The study aimed to understand how the actual activity conducted during the task influenced the learner's metacognitive state. Moreover, based on the literature on personality and individual differences, we investigated how interpersonal traits related to social communication could become predictors for the learner's task activity and his metacognition of it. Using an online tutoring system developed by [11], we collected learners' activity logs of explanations, evaluations of their confidence, and AQ scores. The results of the regression analysis revealed that increasing the amount of actual task work, such as giving many explanations to a social CA, enhances learners' self-confidence about their work, thus supporting hypothesis H1. The analysis of personal characteristics showed that social skills influence self-confidence (thus supporting H2-b); however, they do not influence the actual task work (H2-a is thus not supported). These outcomes indicate that personal traits affect self-confidence in interactions with a social CA. These findings have new implications for designing tutoring systems that can assess and detect learner confidence during online learning activities. An additional analysis using machine learning has also been conducted to investigate if the model suggested in this study could be used to automatically detect learner confidence and thus showed the effectiveness.

Acknowledgments

This work was supported by the Grant-in-Aid for Scientific Research (KAKENHI), No. 15K12410 and 16K00219.

6. REFERENCES

[1] V. A. Aleven and K. R. Koedinger. An effective metacognitive strategy: learning by doing and explaining with a computer-based cognitive tutor. *Cognitive Science*, 26(2):147–179, 2002.

[2] A. Bandura. Self-efficacy mechanism in human agency. *American Psychologist*, 37(2):122–147, 1982.

[3] S. Baron-Cohen, S. Wheelwrite, R. Skinner, J. Martin, and E. Clubley. The autism-spectrum quotient (aq):evidence from asperger syndrome/high-functioning autism, males and females, scientists and mathematicians. *Journal of Autism and Developmental Disorders*, 1(31):5–17, 2001.

[4] A. L. Baylor and Y. Kim. Simulating instructional roles through pedagogical agents. *International Journal of Artificial Intelligence in Education*, 15(1):95–115, 2005.

[5] G. C. 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, Oct 2014.

[6] M. T. Chi, M. Bassok, M. W. Lewis, P. Reimann, and R. Glaser. Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, 13(2):145–182, 1989.

[7] A. H. Danek, T. Fraps, A. von Müller, B. Grothe, and M. Öllinger. Aha! experiences leave a mark: facilitated recall of insight solutions. *Psychological Research*, 77(5):659–669, Sep 2013.

[8] S. D'Mello, S. Craig, A. Witherspoon, B. McDaniel, and A. Graesser. Automatic detection of learner affect from conversational cues. *User Modeling and UserAdapted Interaction*, 18(1):45–80, 2008.

[9] K. W. Eva and G. Regehr. Exploring the divergence between self-assessment and self-monitoring. *Advances in Health Sciences Education*, 16(3):311–329, Aug 2011.

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

[11] Y. Hayashi. Wesca(web-based explanation support with conversational agent): An online learning system to facilitating self-explanations for college studensts. (in preparation).

[12] Y. Hayashi. Influence of social communication skills on collaborative learning with a pedagogical agent: Investigation based on the autism-spectrum quotient. In *Proceedings of the 3rd International Conference on Human-Agent Interaction*, HAI '15, pages 135–138, New York, NY, USA, 2015. ACM.

[13] Y. Hayashi. Lexical network analysis on an online explanation task: Effects of affect and embodiment of a pedagogical agent. *IEICE Transactions on Information and Systems*, E99.D(6):1455–1461, 2016.

[14] K. R. Koedinger, J. R. Anderson, W. H. Hadley, and M. A. Mark. Intelligent Tutoring Goes To School in the Big City. *International Journal of Artificial Intelligence in Education (IJAIED)*, 8:30–43, 1997.

[15] M. Komarraju, K. J. S., R. R. Schmeck, and A. Avdic. The big five personality traits, learning styles, and academic achievement. *Personality and Individual Differences*, 51(4):472 – 477, 2011. Digit Ratio (2D:4D) and Individual Differences Research.

[16] 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.

[17] N. Matsuda, E. Yarzebinski, V. Keiser, R. Raizada, G. J. Stylianides, and K. R. Koedinger. Studying the effect of a competitive game show in a learning by teaching environment. *International Journal of Artificial Intelligence in Education*, 23(1):1–21, 2013.

[18] S. W. Mcquiggan, B. W. Mott, and J. C. Lester. Modeling self-efficacy in intelligent tutoring systems: An inductive approach. *User Modeling and User-Adapted Interaction*, 18(1-2):81–123, Feb. 2008.

[19] J. Metcalfe and D. Wiebe. Intuition in insight and noninsight problem solving. *Memory & Cognition*, 15(3):238–246, 1987.

[20] K. Millis, H. Forsyth, C .and Butler, P. Wallace, A. C. Graesser, and D. Halpern, F. *Serious Games and Edutainment Applications*, chapter Operation ARIES! A serious game for teaching scientific inquiry, pages 169–195. Springer-Verlag, London, 2011.

[21] E. Murphy, I. Grey. M, and R. Honan. Co-operative learning for students with difficulties in learning: a description of models and guidelines for implementation. *British Journal of Special Education*, 32(3):157–164, 2005.

[22] C. Nass, Y. Moon, B. J. Fogg, B. Reeves, and D. C. Dryer. Can computer personalities be human personalities? *International Journal of Human Computer Studies*, 43(2):223–239, 1995.

[23] OECD. *PISA 2015 Results (Volume V): Collaborative Problem Solving.* OECD Publishing, Paris., 2017.

[24] E. Rienovita, M. Taniguchi, M. Kawahara, Y. Hayashi, and Y. Takeuchi. Effect of human agent interaction improves self-esteem and students' motivation. In *Proceedings of the 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI2017)*, pages 673–680. IEEE Computer Society, 2017.

[25] C. Rose and F. Oliver. Technology support for discussion based learning: From computer supported collaborative learning to the future of massive open online courses. 26:660–678, 03 2016.

[26] R. J. Sternberg. *Thinking styles.* New York: Cambridge University Press., 1997.

[27] B. P. Woolf, editor. *Building Intelligent Interactive Tutors.* Morgan Kaufmann, 2009.

Modeling the Effects of Students' Interactions with Immersive Simulations using Markov Switching Systems

Nicholas Hoernle
Harvard University
nhoernle@g.harvard.edu

Kobi Gal
Ben-Gurion University
kobig@bgu.ac.il

Barbara Grosz
Harvard University
barbara@eecs.harvard.edu

Pavlos Protopapas
Harvard University
pavlos@seas.harvard.edu

Andee Rubin
TERC
andee_rubin@terc.edu

ABSTRACT

Simulations that combine real world components with interactive digital media provide a rich setting for students with the potential to assist knowledge building and understanding of complex physical processes. This paper addresses the problem of modeling the effects of multiple students' simultaneous interactions on the complex and exploratory environments such simulations provide. We work towards assisting educators with the difficult task of interpreting student exploration. We represent the system dynamics that result from student actions with a complex time series and use switch based models to decompose the time series into individual periods that target interpretability for teachers. The model learns the transition points between successive periods in the time series as well as the internal dynamics that govern each period. This model differs from other switch based models in that it decomposes the time series in a way that is human interpretable. This approach was applied to data that was obtained from an existing multi-person simulation with pedagogical goals of teaching sustainability and systems thinking. A visualization of the model was designed to validate the interpretability of the generated text-based descriptions when compared to a movie representation of the system dynamics. A pilot study using this visualization indicates that the switch based model finds relevant boundaries between salient periods of student work.

Keywords

Bayesian Inference, Exploratory Learning Environment, Markov Chain Monte Carlo, Interpretability, Switching State Space Models

1. INTRODUCTION

Complex systems simulations are becoming increasingly common in formal and informal STEM learning environments [21]. These simulations present scientific phenomena in a manner that bridges principles of science and the firsthand experience of emergent, real-world outcomes. However, the open-ended and exploratory nature of these simulations presents challenges to teachers' understanding of students' learning. Students' actions have immediate and long-term effects on the simulation leading to a rich array of emergent outcomes. Teachers may wish to discuss students' interactions to highlight salient learning opportunities, but if there are too many "moving parts" to the simulation, this becomes a challenging ideal.

This paper describes an automatic method for extracting salient periods from the log files that are generated by complex exploratory learning environments (ELE). Our goal is to generate relevant summaries of the system dynamics such that teachers can effectively engage students in discussions that stem from their own experiences with the simulations. We study an application of Switching State Space Models (SSSM) to the task of extracting salient periods from a mixed reality ELE, Connected Worlds, installed at the New York Hall of Science (NYSci). SSSMs [7] are a class of model for time-series data where the parameters controlling a linear dynamic system switch according to a discrete latent process. These models have seen use in a wide variety of domains including control [11], statistics [2], econometrics [8] and signal processing [14]. SSSMs combine hidden Markov and state space models to capture *regime* switching in non-linear, continuous valued time series [22]. The intuition is that a system evolves over time but may undergo a regime change that results in an intrinsic shift in the system's characteristics. Allowing for discrete points in time where the dynamics change, enhances the power of the simple linear models to capture more complicated dynamics. We propose that regime switching models also help to increase the *interpretability* of large and complex systems by automatically segmenting a time series into regions of approximately uniform dynamics. The result is that a complex session is broken into smaller periods that are more readily understood upon reflection on the session.

In this paper we introduce the Connected Worlds ELE and explain why teachers might need assistance when leading a discussion with the students where they reflect upon their actions. We expound on the SSSM and propose a method for decomposing a complex time series into smaller periods aiming to assist teachers when reflecting on a session with a class. We lastly present results showing the efficacy of

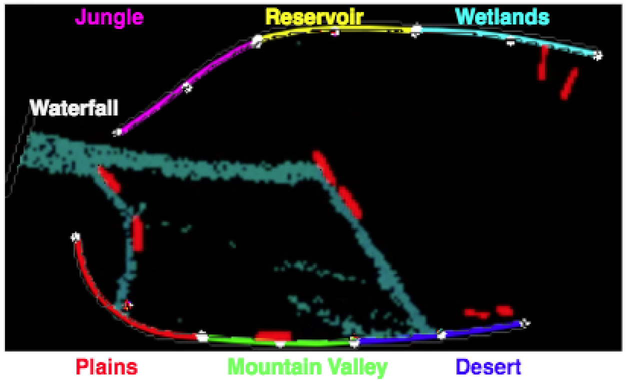


Figure 1: Bird's eye snapshot view taken from the movie representation of the CW environment. Biomes are labeled on the perimeter and logs appear as thick red lines. Water enters via the waterfall and in this image it mainly flows toward the desert and the plains.

our approach on both synthetic data and on data collected from CW. The CW validation is a preliminary study with significant results which suggest that the model output is human interpretable.

2. CONNECTED WORLDS

Connected Worlds[1] (CW) is a multi-person ecology simulation with the goal of teaching students about complex systems and systems thinking. It consists of an immersive environment comprising four interconnected biomes connected by a central flow of water that is fed by a waterfall. The simulation exhibits large scale feedback loops and presents the opportunity for participants to experience how their actions can have (often unintended) effects that are significantly removed in time and-or space. Students plant trees which flourish or die, animals arrive or depart, and rain clouds form, move through the sky and deposit rain into the waterfall.

Students interact with CW by positioning logs to control the direction of the water that flows in the simulation. Water can be directed to each of the four biomes (desert, plains, jungle, wetlands) and the distribution of flowing water depends on the placement of the logs. Water enters the simulation in two ways. The students can actively release water into the system from the stored water in the reservoir. Rainfall events are out of the students' control and these release water into the waterfall (to replenish the primary source of water) and into the individual biomes.

Figure 1 shows a bird's eye snapshot view of the state of the simulation in CW. The nature of the simulation is complex on a variety of dimensions. The simulation involves a large number of students simultaneously executing actions that change the state of the simulated environment. No one person - including the teacher or interpreter - can possibly follow what happens, even in a relatively short simulation. Each participant will have a different view of what transpired, depending on the actions s/he took and the state changes that resulted. Thus it is important to develop tools that can support teachers' understanding of the effects of students' interactions in complex ELEs such as CW.

3. RELATED WORK

This work is related to two separate strands of research: studying students' interactions in mixed reality ELEs, and modeling complex systems using switching models.

There is increasing evidence of the value of multi-person participatory simulations for engaging learners with complex science topics [9, 1, 23]. Research has explored classroom-scale participatory simulations where students play active roles in the simulation. Some examples include topics in disease transmission [3] and human body systems [12]. Other work has placed students in the role of scientists experimenting with simulated ecosystems [17, 4]. Within all of these examples, learners both engaged directly with the simulation during enactment, and reflected on their actions afterward to better understand how their choices resulted in observed system outcomes. Research has shown that using data obtained from students' own performances has the potential to engage them more effectively than presenting them with the results of an abstract simulation [16, 15]. Building on this work, our eventual goal is to provide assistive tools for teachers to further enhance the pedagogical impact that such ELEs can achieve.

Much work has been completed in the field of mining meaningful knowledge from time series data [5, 10, 19]. Ghahramani and Hinton [7] introduce and give a detailed presentation of the SSSM. We adapt this model to the special structure that is inherent in CW. Cappé et al. [2] and Giordani et al. [8] use switching models to capture non-linear behavior in a time series. SSSMs have been effectively applied in object tracking domains where it is necessary to predict the trajectory of various objects. Whiteley et al. [22] introduce a sequential Monte Carlo algorithm for inference over switching state space models using discrete particle filters. We present a new avenue of study in which SSSM models are used to describe complex time series in a way that can be easily interpreted by people.

4. SWITCHING STATE SPACE MODELS

SSSMs are commonly used to describe time series[2] with nonlinear dynamics in econometrics and signal processing applications [8, 14]. A SSSM includes M latent continuous valued state space models and a discrete valued switching variable. Each of the models, which we refer to as regimes, have their own dynamics. At each point in time, the switching variable selects one of the individual state-space models to generate an observation vector.

The SSSM is formalized as:

$$\begin{aligned} \mathbf{X_t^{(m)}} &= \Phi^{(m)}\mathbf{X_{t-1}^{(m)}} + w_t^{(m)} \\ \mathbf{Y_t} &= S_t A^{(m)}\mathbf{X_t^{(m)}} + v_t \end{aligned} \tag{1}$$

Here, $X_t^{(m)}$ denotes the latent continuous valued state for

[1] https://nysci.org/home/exhibits/connected-worlds/

[2] Refer to Shumway and Stoffer [20] for a detailed discussion of time series analysis models.

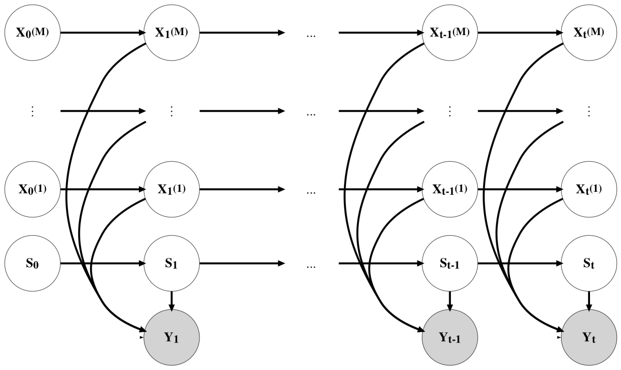


Figure 2: Graphical model for the switching-state space model. A latent discrete switching variable (S_t) selects an active, continuous state space model ($X_t^{(m)}$). The observation vector (Y_t) depends on the active regime at time t.

regime m at time t. S_t is a switching variable that selects the m^{th} regime such that regime m at time t produces observation vector Y_t, which depends on the latent state $X_t^{(m)}$. The states $X_t^{(m)}$ evolve over time in a way that depends on the transition matrix $\Phi^{(m)}$ and the previous state X_{t-1}. Figure 2 presents a graphical representation of an SSSM. Edges between variables represent stochastic casual relationships. Not shown in the figure are the regime dependent transition noise $w_t^{(m)}$ and the observation noise v_t. $A^{(m)}$ is the output matrix in the state space formulation, set to identity matrix I in our case.

We illustrate how an SSSM can describe the effects of students' interactions in CW. Y_t represents the observed water level in the different areas of the simulation at time t. $X_t^{(m)}$ describes the expected levels of water under regime m at time t. $\Phi^{(m)}$ controls the water flow in the simulation according to the transitions in regime m. S_t selects which of the regimes to use to describe the water level Y_t.

Importantly, a single regime is insufficient for modeling the effects of students' interactions with CW. This is because students' actions have a complex impact on the system dynamics. We therefore need to define multiple regimes, where each regime describes a series of events that can be (stochastically) explained by the regime dynamics. A regime is active for a duration of time in CW; we call this duration a period. For example, in one period water is mainly flowing to the plains and to the desert (as is shown in figure 1). In the next period, students move the logs to re-route water flow to the wetlands potentially because plant life is dying. Each of these periods might be active for different durations and their dynamics are described by different regimes.

4.1 Exploiting Model Structure

We aim to perform inference over the latent states, $X_t^{(m)}$, the regime parameters, $\Phi^{(m)}$, and latent switching variable, S_t. Computing posterior distributions for SSSM is computationally intractable [18]. To illustrate, in figure 2 we see that the graph consists of M state space models that are marginally independent. These models become conditionally dependent when Y_t is observed, as is the case in this graph. The result is that $X_t^{(m)}$ is conditionally dependent on the value of all of the other latent states and switching variables for times 1 through T and regimes 1 through M [18]. Previous approaches use approximate methods such as variational inference [7] and a 'merging of Gaussians' [14, 18] to address the inference problem. The variational inference approximation transforms the intractable Bayesian expectation problem into an optimization problem by minimizing the Kullback Leibler (KL) divergence between a simpler family of approximating distributions and the unknown, intractable posterior. The merging of Gaussians approach uses a single Gaussian to represent the mixture of M Gaussians at each time step thereby simplifying the computation with the cost of being susceptible to local optima (see section 5.1).

While these methods have seen success in previous examples, they cannot be applied to our domain. This is because they allow the system to switch back and forth between regimes, resulting in frequent regime changes that can hinder the interpretability of the model output. This work takes a different approach by imposing structure on the model to address both inference and interpretability challenges. Further, as the optimization procedures of the previous work are susceptible to local optima, we rather use a Markov chain Monte Carlo (MCMC) approach to approximate the posterior distribution of the latent parameters.

We make two assumptions, which arise from the need to create human interpretable descriptions of complex system behavior. **Assumption 1:** the system advances through a series of regimes, each regime is active for a period, and then switches to an entirely new regime, one that has not been used before. **Assumption 2:** the regime remains active for the maximum possible time for which it can be used to describe the period.

To illustrate, without making these assumptions there are M possible assignments of regimes for each time step, making a total of M^T combinations of possible assignments, which is exponential in the number of time steps. Moreover, in the worst case, the number of possible periods is bounded by T with a switch at every time step. In contrast, under our assumptions, there are only two possible assignments of regimes for each time step (i.e., do we stay in the current regime or do we progress to the next regime), making for a total of 2^M combinations of possible assignments, where M is constant. The number of possible periods under this methodology is bounded by M. We hypothesize that the forced reduction in complexity of the fitted model would significantly simplify the interpretability of the model for a human.

4.2 Algorithm for Posterior Inference

Computing the posteriors in an SSSM corresponds to approximating the joint distributions over $X_t^{(m)}$ and $\Phi^{(m)}$ given the observation vector $\mathbf{Y}$. A well known problem with MCMC inference in complex graphical models with hidden variables is that of identifiability [13]. Models are nonidentifiable when two sets of parameters can explain the observed data equally well. For example, in a simple Gaussian mixture model with means μ_0, μ_1 and covariances Σ_0, Σ_1, the marginal posterior distributions of the parameters are identical. A possible solution to the identifiability problem is to add constraints (e.g. enforcing $\mu_0 > \mu_1$). However, defining constraints in higher-dimensional domains is non-trivial.

Another solution for solving the identifiability problem is to provide labels for part of the data. This is termed semi-supervised learning and we incorporate this solution into our model. In the context of the CW domain, we can label observations as belonging to one regime or another. Let $S_{t,t+1,...,t-1+K,t+K}$ be a consecutive set of K state variables such that S_t and S_{t+K} have known value assignments (regime m and regime $m+1$ respectively). The values for the state variables $S_{t+1,...,t-1+K}$ are unknown. By Assumption 1, the switch between regimes m and $m+1$ occurs at some S_l where $t < l \leq t+K$. Therefore, the value of S_l determines the values for all of the unknown states as S_t is assigned to regime m for $t < l$ and it is assigned to regime $m+1$ for $t \geq l$.

We provide a sketch of this process in Algorithm 1. Step 1 initializes the M supervised switch variables, one per regime. The labeled switch variables are spaced uniformly in time and are assigned to regimes in increasing order according to Assumption 1. This uniform method for initialization can be justified by Assumption 2, in that any set of regimes that provides an interpretable model is sufficient. The number of expected time steps in each period is $K = T/M$, and there are $K-2$ unlabeled switch variables between each pair of switch variables assigned to regimes.

Step 2 performs MCMC sampling to approximate the posterior of the model[3]. For the case when the value of the switch variable is known, the posterior of $X_t^{(m)}$ can be directly sampled by following the structure of a state space model. In the case where the switch variable is unknown, we have a marginalization problem over the two possible values of S_t. For the hidden Markov model (HMM) structure this can be efficiently computed with the forward algorithm [20]. To formulate the HMM forward algorithm, we use the observation probabilities from the individual state space models in place of the emission probabilities of a standard HMM. Here, π_{S_i} refers to the belief of the state of the switching variable given the evidence up to that point in time.

Step 3 uses the regime specific parameters $\Phi^{(m)}$ to make a maximum likelihood assignment of an observation to a regime using the Viterbi algorithm [20], thereby specifying the value of $S_t \forall t \in [1:T]$.

Algorithm 1 is computed on an SSSM that implements Assumptions 1 and 2. Such a model is shown in figure 3. The model depicts a subset of the time series with K time steps from time t to time $t+K$. There are two supervised labels at the boundaries of the subset with the variable S_t assigned to regime m and variable S_{t+K} assigned to regime $m+1$. The unknown $K-2$ states in between are marginalized over such that the regime specific posteriors can still be approximated. This model is repeated for the $M-1$ switches in the data. The setup is flexible in that informative priors for the model noise and transition matrices can be specified (and

Algorithm 1: Posterior inference algorithm

```
Input: $M$ (number of regimes), $\mathbf{Y}$ (vector of observations
  for $T$ time steps).
1 Initialization: Label one datapoint per regime, leaving
    $T-(M+1)$ unlabeled datapoints.
2 MCMC Inference: Draw samples for $X_t^{(m)}, \Phi^{(m)}$ from the
    posterior distribution defined by the structured
    probability model:
  for $Y_t$ in $\mathbf{Y}$ do
      if $S_t = m$ is known then
          sample from $P(X_t^{(m)}, \Phi^{(m)} \mid X_{t-1}, S_t = m, Y_t)$
      else
          marginalize over $S_t$. Sample from
            $\sum_{i=m-1}^{m} \pi_{S_i} P(X_t^{(i)}, \Phi^{(i)} \mid X_{t-1}, S_t = i, Y_t)$
3 Posterior Inference: Use the posterior for regime
    parameters ($\Phi^{(m)}$) to run a Viterbi pass on the data $\mathbf{Y}$ to
    make a maximum likelihood assignment of the value of $S_t$
    to regime $m$ (thereby learning the switching variables $S_t$).
Output: $S_t$ (assignments to regimes), $\Phi^{(m)}$ (regime
  posterior distributions).
```

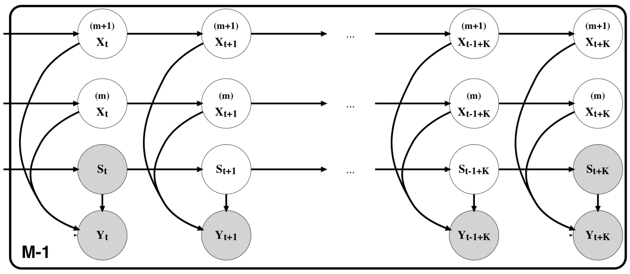


Figure 3: Updated graphical model showing the semi-supervised switching labels, along with the choice of only two chains between two semi-supervised points. This representation is repeated $M-1$ times to describe the $M-1$ switches between the M regimes.

[3] Implemented using Stan MC (http://mc-stan.org/)

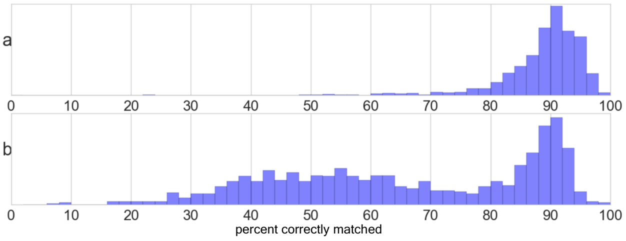


Figure 4: Histogram of the percent of correctly inferred labels for the observed output. The structured sampling Algorithm 1 (a) learns the regime labels more accurately than the randomly initialized Gaussian merging algorithm (b).

related) as required by domain knowledge.

5. EMPIRICAL VALIDATION

We evaluate two aspects of Algorithm 1. First, we show that it finds the true regime labels in a synthetic dataset. Thereafter, we use data that were collected from Connected Worlds to run a preliminary experiment that tests whether the inferred periods are interpretable to human validators.

5.1 Evaluation on Synthetic Data

We generate synthetic data to test whether Algorithm 1 finds a reasonable representation of known switches in an SSSM. Equation 2 describes an SSSM with two regimes and a continuous state space. The transition parameters and regime noise are determined according to the active regime. This model is adapted from Ghahramani and Hinton [7] which describes a state space that is disjoint at regime switches; we rather chose to make the state space continuous at the switch points as this more accurately mimics the scenario that is present in CW. The prior probability of each of the regimes is 0.5 ($p_1 = p_2 = 0.5$); the regime transition probabilities are $S_{1,1} = S_{2,2} = 0.95$ and $S_{1,2} = S_{2,1} = 0.05$[4]. We used this model to generate 1000 time series, each with 200 observations.

$$\begin{aligned}\mathbf{X_t^{(1)}} &= 0.99\,\mathbf{X_{t-1}} + w_t^{(1)} \quad & w_t^{(1)} &\sim \mathcal{N}(0,1)\\ \mathbf{X_t^{(2)}} &= 0.9\,\mathbf{X_{t-1}} + w_t^{(2)} \quad & w_t^{(2)} &\sim \mathcal{N}(0,10)\\ \mathbf{Y_t} &= S_t\mathbf{X_t} + v_t \quad & v_t &\sim \mathcal{N}(0,0.1)\end{aligned} \tag{2}$$

We compare the Gaussian merging baseline that is commonly used in the literature [14] to Algorithm 1 with the number of regimes initialized to 9. The accuracy of each approach is measured as the percentage of the correctly labeled data points as belonging to either regime 1 or regime 2. On average Algorithm 1 labels 89% of the data correctly, materially higher than the 66% average accuracy of the Gaussian merging approach. Figure 4 shows a histogram of the correctly inferred switch points in the data according to Algorithm 1 (top) and the baseline (bottom). The bi-modal and long tailed distribution for the baseline approach demonstrates its susceptibility to local optima.

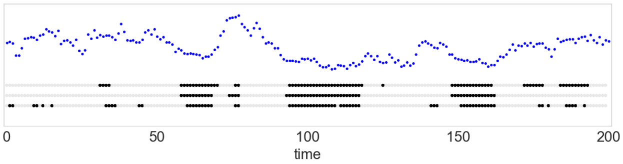


Figure 5: An example of a generated time series from the SSSM model of Equation 2. The x axis represents time, and the y axis shows the observations (the magnitudes of the signal are irrelevant for this investigation). Regime labels are shown as black and gray dots representing the two label options. True labels (top) are compared to the inferred labels from Algorithm 1 (middle) and the Gaussian merging (bottom).

Figure 5 shows an example of the generated time series (top) and the associated switch points (bottom). The switch points are shown according to the true model, the points inferred by Algorithm 1 and the points inferred by the baseline. Each period is represented by a sequence of black and gray colored circles. As shown by the figure, the periods inferred by Algorithm 1 and the baseline both overlap to some extent with the true periods. However, there is substantially more noise in the inferred periods of the baseline. Algorithm 1 learns the regime autoregressive parameters $\phi_1 = 0.97 \pm 0.027$ and $\phi_2 = 0.88 \pm 0.035$, again showing an effective recovery of the individual regime parameters.

The superior performance of Algorithm 1 can be directly attributed to the switching behavior that is enforced by Assumptions 1 and 2, which was not assumed by the baseline model. Although the model structure encourages the discovery of switches in Algorithm 1 the uniformly spaced labels should not be seen as a model advantage as no prior knowledge of the actual switches is used in performing this initialization step. Given that the proposed algorithm finds a reasonable representation for the switches in a generated dataset, we turn to the evaluation of the interpretability of its output within the CW context.

5.2 Preliminary Validation of Interpretability on Connected Worlds Data

Because the ultimate users of the output of Algorithm 1 will be teachers leading their students in a discussion of the simulation behavior, we wanted to confirm that the inferred switch points were interpretable by a human seeking to understand the “story” of the simulation. In order to do this, we used a movie of the water flow (see figure 1 for one such frame) and asked evaluators to select one of three possible switch points between every pair of consecutive periods. Evaluators saw a composite of 1) the movie of the two periods; 2) a description of the dynamics of each of the two periods and 3) a set of three possible switch points between the periods. The evaluator’s task was to choose the switch point that best matched the change in dynamics between the two periods. One of the three switch points was that inferred by Algorithm 1; the other two were random times sampled uniformly from the beginning of the first period to

[4] $S_{j,k}$ denotes the probability of a switch from regime j to regime k.

the end of the second period.[5]

The descriptions were generated from the inferred parameters that are an output from Algorithm 1. In equation 1, $\Phi^{(m)}$ refers to the transition matrix for the m^{th} regime. As is discussed in section 4, the parameters from this matrix describe the expected movement of water in the given period. We threshold the values from this matrix to generate a short text description for the water movement. One such description could be: *"Water is directed towards the desert and plains. The wetlands and jungle are receiving little or no water"*.

Evaluators worked with five sessions, each of which included 5 to 10 periods of system dynamics. Selecting the correct switch point is not a trivial task: it requires distinguishing between changes in the system that indicate different dynamic regimes and those that are noise within the same dynamic regime. We see an evaluator's ability to choose a switch point based on the movie and a description of the two contiguous periods as evidence that the inferred periods are usable by a teacher who wants to guide students in constructing a causal description of their experience with the simulation. Moreover, this can be seen as evidence that the inferred regime parameters match inferred period boundaries, together presenting a coherent description for the water movement for a short segment of the CW session.

Figure 6 shows the results of the validation using four evaluators with knowledge of the CW domain. The five sessions are shown along the x-axis; the fraction of correctly selected switch points is shown by the bin heights. The dashed line represents a random baseline in which the selected switch probability corresponds to $\frac{1}{3}$. Under the null hypothesis, the performance of an evaluator would not be significantly different than the random baseline. The results indicate that the evaluators chose the switch point identified by Algorithm 1 significantly more often than the random baseline ($p < 1 \times 10^{-4}$), suggesting that the inferred switch points were indeed interpretable to a large extent as meaningful changes in the state of the system. The differences in interpretability seen in figure 6 (e.g. session 4 was more difficult to interpret than session 3) can provide further guidance to us in how to support teachers and students in making sense of their experiences in CW. For example, the sessions with more complicated dynamics might need more periods to fully capture the progression over time. Predefining the number of periods for a given session is an aspect of this approach that needs addressing. A more detailed user study is left for future work.

6. CONCLUSION AND FUTURE WORK

This paper has presented novel research into the simplification of log files that are generated by complex participatory immersive simulations. The log files were represented as a time series that was decomposed with the long term goal of producing periods that are useful for a teacher when leading reflective discussions about students' sessions. We have built upon previous time series analysis tools to formulate a model that automatically segments a time series into these salient periods. The efficacy of the algorithm was demonstrated on a synthetic dataset where it outperformed previous work at the task of assigning data to regimes. We used the algorithm's output to generate a short text description of the dynamics in an inferred period. We find that evaluators are independently able to validate the inferred changes between the automatically generated periods. This preliminary study demonstrates that it is possible to simplify a time series log into periods of activity that are human interpretable.

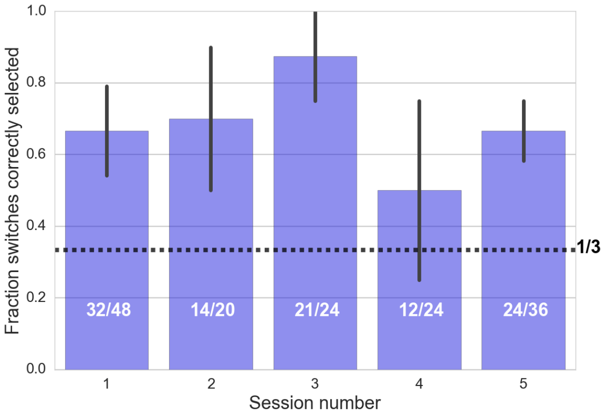


Figure 6: Expert validation of five different test files from sessions with CW. The histogram shows the fraction of correctly identified switches between automatically identified periods with an expected baseline accuracy of $\frac{1}{3}$.

Our focus now rests on designing assistive tools for teachers that can facilitate their understanding of students' interactions in multi-participant immersive simulations. Moreover, our results suggest that the model should be capable of adapting the number of inferred regimes to the complexity of a given session. Fox et al. [6] explore a Bayesian non-parametric model which allows the data to dictate the number of regimes that are inferred. The application of this model to the CW data presents an attractive tool for remaining agnostic about the number of regimes that are present in a session. Another avenue for future research involves exploring the trade-off that is made between the predictive power of a model and the explanatory coherence that the model achieves. Wu et al. [24] have suggested a method for regularizing deep learning models to facilitate people's understanding of their predictions. This is an important balance to consider and one that we intend to consider in educational settings.

7. ACKNOWLEDGMENTS

This paper is based on work supported by the National Science Foundation under grant IIS-1623124. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Thank you to Prof. Leilah Lyons (NYSci; University of Illinois-Chicago) and Aditi Mallavarapu (University of Illinois-Chicago) for their input on this work; and NYSci for assisting with the data collection.

[5] Visualization available at `https://s3.amazonaws.com/essil-validation/index.html`.

8. REFERENCES

[1] C. Brady, M. Horn, U. Wilensky, A. Wagh, A. Hjorth, and A. Banerjee. Getting your drift–activity designs for grappling with evolution. In *Proceedings of International Conference of the Learning Sciences, ICLS*, volume 3, pages 1603–1604. International Society of the Learning Sciences, 2014.

[2] O. Cappé, E. Moulines, and T. Rydén. Inference in hidden markov models. In *Proceedings of EUSFLAT Conference*, pages 14–16, 2009.

[3] V. Colella. Participatory simulations: Building collaborative understanding through immersive dynamic modeling. *The journal of the Learning Sciences*, 9(4):471–500, 2000.

[4] C. Dede, T. A. Grotzer, A. Kamarainen, and S. J. Metcalf. Virtual reality as an immersive medium for authentic simulations. In *Virtual, Augmented, and Mixed Realities in Education*, pages 133–156. Springer, 2017.

[5] P. Esling and C. Agon. Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1):12, 2012.

[6] E. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. Nonparametric bayesian learning of switching linear dynamical systems. In *Advances in Neural Information Processing Systems*, pages 457–464, 2009.

[7] Z. Ghahramani and G. E. Hinton. Variational learning for switching state-space models. *Neural computation*, 12(4):831–864, 2000.

[8] P. Giordani, R. Kohn, and D. van Dijk. A unified approach to nonlinearity, structural change, and outliers. *Journal of Econometrics*, 137(1):112–133, 2007.

[9] A. Gnoli, A. Perritano, P. Guerra, B. Lopez, J. Brown, and T. Moher. Back to the future: embodied classroom simulations of animal foraging. In *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction*, pages 275–282. ACM, 2014.

[10] B. Horst and K. Abraham. *Data mining in time series databases*, volume 57. World scientific, 2004.

[11] N. Ikoma, T. Higuchi, and H. Maeda. Tracking of maneuvering target by using switching structure and heavy-tailed distribution with particle filter method. In *Control Applications, 2002. Proceedings of the 2002 International Conference on*, volume 2, pages 1282–1287. IEEE, 2002.

[12] A. Ioannidou, A. Repenning, D. Webb, D. Keyser, L. Luhn, and C. Daetwyler. Mr. vetro: A collective simulation for teaching health science. *International Journal of Computer-Supported Collaborative Learning*, 5(2):141–166, 2010.

[13] A. Jasra, C. C. Holmes, and D. A. Stephens. Markov chain monte carlo methods and the label switching problem in bayesian mixture modeling. *Statistical Science*, pages 50–67, 2005.

[14] C.-J. Kim, C. R. Nelson, et al. State-space models with regime switching: classical and gibbs-sampling approaches with applications. *MIT Press Books*, 1, 1999.

[15] V. R. Lee and J. Drake. Quantified recess: design of an activity for elementary students involving analyses of their own movement data. In *Proceedings of the 12th international conference on interaction design and children*, pages 273–276. ACM, 2013.

[16] V. R. Lee and J. M. Thomas. Integrating physical activity data technologies into elementary school classrooms. *Educational Technology Research and Development*, 59(6):865–884, 2011.

[17] T. Moher, B. Uphoff, D. Bhatt, B. López Silva, and P. Malcolm. Wallcology: Designing interaction affordances for learner engagement in authentic science inquiry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 163–172. ACM, 2008.

[18] K. P. Murphy and S. Russell. Dynamic bayesian networks: representation, inference and learning. 2002.

[19] P. Patel, E. Keogh, J. Lin, and S. Lonardi. Mining motifs in massive time series databases. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 370–377. IEEE, 2002.

[20] R. H. Shumway and D. S. Stoffer. Time series analysis and its applications. *Studies In Informatics And Control*, 9(4):375–376, 2000.

[21] O. Smørdal, J. Slotta, T. Moher, M. Lui, and A. Jornet. Hybrid spaces for science learning: New demands and opportunities for research. In *International Conference of the Learning Sciences. Sydney, Australia.*

[22] N. Whiteley, C. Andrieu, and A. Doucet. Efficient bayesian inference for switching state-space models using discrete particle markov chain monte carlo methods. *arXiv preprint arXiv:1011.2437*, 2010.

[23] U. Wilensky and M. Resnick. Thinking in levels: A dynamic systems approach to making sense of the world. *Journal of Science Education and technology*, 8(1):3–19, 1999.

[24] M. Wu, M. C. Hughes, S. Parbhoo, M. Zazzi, V. Roth, and F. Doshi-Velez. Beyond sparsity: Tree regularization of deep models for interpretability. *arXiv preprint arXiv:1711.06178*, 2017.

Using a Common Sense Knowledge Base to Auto Generate Multi-Dimensional Vocabulary Assessments

Ruhi Sharma Mittal
IBM Research
Bangalore, India
ruhi.sharma@in.ibm.com

Seema Nagar
IBM Research
Bangalore, India
senagar3@in.ibm.com

Mourvi Sharma
IBM Research
Bangalore, India
mourshar@in.ibm.com

Utkarsh Dwivedi[*]
StudyPad
Bangalore, India
dwivediu@acm.org

Prasenjit Dey
IBM Research
Bangalore, India
prasenjit.dey@in.ibm.com

Ravi Kokku
IBM Research
Yorktown, US
rkokku@us.ibm.com

ABSTRACT

As education gets increasingly digitized, and intelligent tutoring systems gain commercial prominence, scalable assessment generation mechanisms become a critical requirement for enabling increased learning outcomes. Assessments provide a way to measure learners' level of understanding and difficulty, and personalize their learning. There have been separate efforts in different areas to solve this by looking at different parts of the problem. This paper is a first effort to bring together techniques from diverse areas such as knowledge representation and reasoning, machine learning, inference on graphs, and pedagogy to generate automated assessments at scale. In this paper, we specifically address the problem of Multiple Choice Question (*MCQ*) generation for vocabulary learning assessments, specially catered to young learners (YL). We evaluate the efficacy of our approach by asking human annotators to annotate the questions generated by the system based on relevance. We also compare our approach with one baseline model and report high usability of *MCQs* generated by our system compared to the baseline.

Keywords

Knowledge base, Vocabulary learning, Vocabulary Assessment, Assessment Generation, MCQ Generation, Personalized Vocabulary learning

1. INTRODUCTION

Personalized automated tutoring provides a scalable solution for augmenting in-class learning, and hence helps teachers effectively achieve increased learning outcomes in multi-student classrooms.

In automated tutoring, assessments play an important role, since they provide a way to continuously measure learners' level of understanding for a given concept. For young children, automatic vocabulary assessment is an interesting research problem and several efforts have been devoted to it [9, 17, 18, 23, 29]. It is a complex problem since word knowledge acquisition for first language learners is an incremental, continuous process, in part determined by the richness of a word's connection to other related words [5, 8]. This is important because the more associations a word has, the easier it is for learners to identify the meaning of the word when it is encountered again in a new context [7]. Hence, automatic assessment generation should strive to assess the multiple facets of a word, in the context of its relationships with other words.

Among the different assessment types, an *MCQ* test is a simple and highly scalable assessment mechanism, and is easily gamifiable for increased engagement by young learners. In this paper, we mainly focus on *MCQ* generation with a single correct answer and multiple distractors, although the solution is equally and trivially extensible to *MCQs* with multiple correct answers. There are three important parts of an *MCQ*, a) a Question Stem, b) a Correct Answer and c) one or more Distractors. For a young language learner, the scope of varying the question stem and the correct answer is limited, but distractors play an important role in determining the level and relevance of an automatically generated *MCQ*. Generating the right set of distractors for an *MCQ* is a difficult and tedious task even for humans. Hence, our main attention in this paper is on automatic generation of good distractors for *MCQs*.

We use ConceptNet5.4 [19] as a common sense knowledge base (KB) and generate a diverse set of *MCQs* for assessing conceptual understanding of a word. Using ConceptNet, however, leads to several challenges: 1) some of the links may not be appropriate to vocabulary learning for young learners, 2) there may be missing or sparse links for some words, and 3) there is no explicit information about word sense. To address these challenges, we first employ a supervised learning approach to filter out inappropriate links before generating *MCQs*. Second, we employ word embeddings [28] to overcome missing and sparse links. Third, even

[*] Work done while at IBM Research.

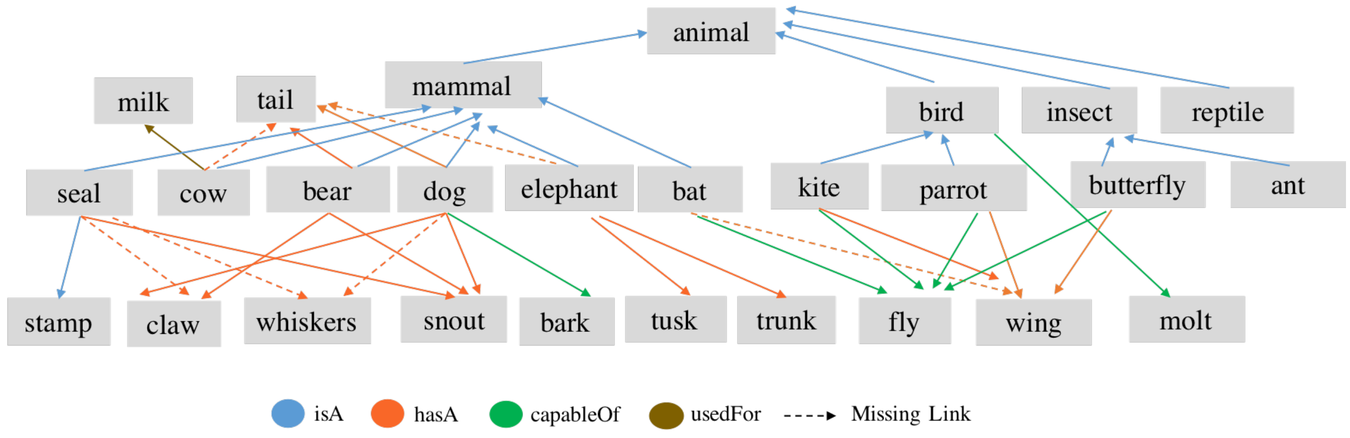


Figure 1: Snapshot of YL-KB

though information on multiple meanings of the same word is not directly available in ConceptNet, we detect the presence of multiple meanings of words through varying independent relationships in ConceptNet based graph (e.g. seal–the animal and seal–the stamp have independent hierarchies of word relationships in the graph), and hence are able to generate questions which aim to assess knowledge of multiple meanings of a word. Hereafter, we refer to this curated ConceptNet for young learners as YL-KB (Young Learners Knowledge Base).

We evaluate the efficacy of our approach by asking human annotators to annotate the questions generated by the system based on the relevance and the automatic difficulty level assigned. We also compare our approach with two baseline models. We perform extensive evaluation on a set of 600 automatically generated questions. For relevance of the generated *MCQs* we report Fleiss Kappa [14] *moderate* (0.44) inter-annotator agreement.

The paper is organized as follows. We review the related work on question generation as it applies to *MCQs* in Section 2. We describe our design considerations, and approach along with the system architecture in Sections 3 and 4 respectively. We report the results of our evaluation in Section 5 and conclude in Section 6.

2. RELATED WORK

Prior research has mainly addressed *MCQ* generation from two dimensions, namely 1) utilizing text corpora and lexical resources such as WordNet [13] to generate question stem, correct answers and distractors, and 2) utilizing domain ontologies to generate domain specific *MCQs*. Some notable work utilizing WordNet[13] lexical resource for generating MCQs are [9, 17, 20, 18]. Brown et al. [9] generate different types of questions for a word, aiming to assess different aspects such as synonyms, antonyms, definition etc. The approach for choosing distractors is to pick words which have the same part of speech as the word in the question stem. Hoshino et al. [17] present different methods for generating distractors, such as mutual information and edit distance, using WordNet. Mitkov et al. [20] find keywords based on frequency of occurrences and create a question for a word based on the phrase it is occurring in. They use WordNet's hypernym relationship to find distractors. Generation of *MCQ* distractors using WordNet for English language adjective understanding is discussed in [18]. Gates et al. [15] use definitions for words to generate a cloze type question for vocabulary building. They remove verb phrase to create cloze type question. For distractor generation, they employ a simple technique where phrases which have same structure as the answer phase are the potential distractors. Mostow et al. [21] propose automatic generation of multiple choice cloze questions to test a child's comprehension while reading a given text. Unlike previous methods, it generates different types of distractors designed to diagnose different types of comprehension failure, and tests comprehension not only of an individual sentence but of the context that precedes it. More recent work aims to generate *MCQs* for any Wikipedia topic [16] and using DBpedia [27] fills the gap of generating *MCQs* for quiz-style knowledge questions from a knowledge graph such as DBpedia[6].

A number of papers utilize domain ontology for automatic question generation. Some notable works in this domain are [24, 3, 1, 4, 12, 2, 30], which address different aspects of automatic question generation from domain based ontology: 1) how to generate distractors; 2) how to control the difficulty of a question; 3) how to control the number of questions to be generated, since in a practical setting only a specific number of questions would make sense; 4) how to generate domain relevant questions and 5) limitations of using domain ontology for automatic question generation. Our paper advances the state-of-the-art in significant ways. It cuts across all different dimensions of generating *MCQs* for assessing vocabulary learning in *young* children, by using a common sense knowledge base with wider coverage but high noise. To the best of our knowledge, this is the first of its kind work that addresses the sophisticated task of automatically generating varying word knowledge assessments using techniques from diverse areas of knowledge representation and reasoning, machine learning, inference on graphs, and pedagogy. Further, using ConceptNet instead of WordNet provides significant advantages in terms of the number and diversity of word relationships available.

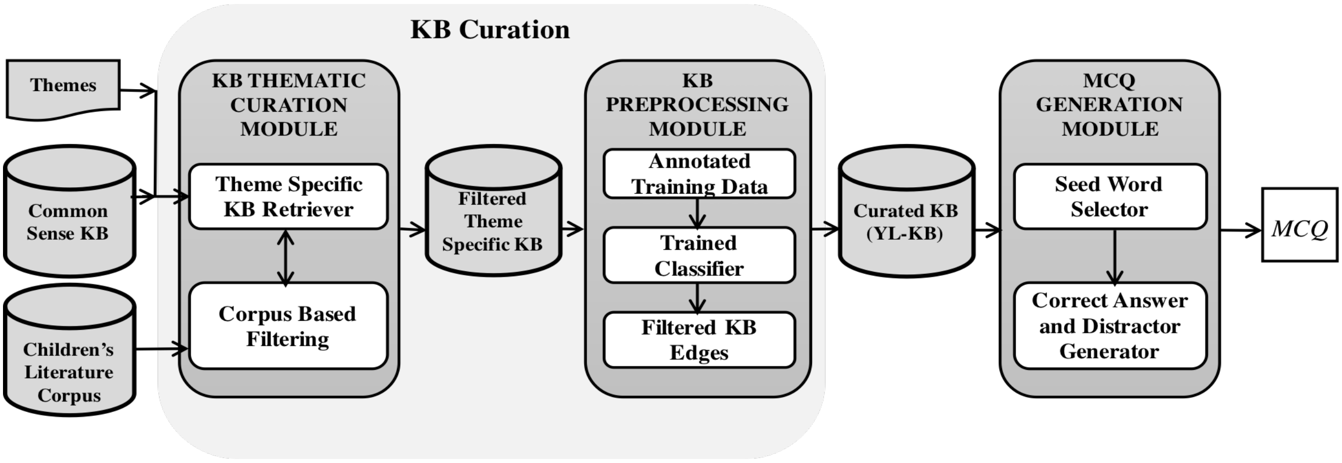


Figure 2: High Level Solution Overview

3. DESIGN CONSIDERATIONS

Our approach to *MCQ* generation builds on ConceptNet, a semantic network containing common sense knowledge (often stored as networks of related ideas) created to help computers understand the world. When a learner is assimilating information about words in a language, in a way they are trying to make a mental semantic network of words [22]. Since a common sense knowledge base mirrors this semantic network, it can potentially serve as a resource for generating robust, multi-dimensional assessments for vocabulary learning.

4. OUR APPROACH

In this section, we first present the definitions of terms we use throughout the paper. Then, we present the high level overview of our system for automatic *MCQ* generation from a common sense KB. A detailed explanation of each component of the system follows in further subsections.

4.1 Terminology

A common sense knowledge base (KB) is a directed semantic network of common sense entities such as words and phrases as shown in Figure 1. The entities in the network are connected with a diverse set of semantic relations such as *isA*, *hasA*, *atLocation* etc. We represent the semantic network as graph G, consisting of V nodes and E edges, where edge labels come from the conceptual relationship connecting the two nodes in a directed manner. We call relations representing functional characteristics such as *hasA*, *usedFor*, and *capableOf* functional relations, and *isA* as hierarchical relation. We represent the words and relations for which we create *MCQs* as seed words and seed relations, respectively. If there is a directed edge from node c to node p with relation r, we call p as a parent of node c with relation r and c as a child of node p with relation r. The siblings of a node, with respect to a specific relation, are defined as all the children of its parent node, except for the node itself.

4.2 System Architecture

Our goal is to enable a holistic solution for automatic *MCQ* generation from a KB. The high level overview of our solution is depicted in Figure 2. The KB is curated for themes that are relevant for young learners and then filtered using the Children's Book Test [26] corpus, which is a dataset curated from an extensive selection of children's books. Further filtering is done to remove noisy and irrelevant edges. The curated KB, referred to as YL-KB is now free from inappropriate and noisy data, which makes it suitable to use for vocabulary assessments. The YL-KB is used to select seed words and generate all six types of questions. We now discuss each of these stages in detail.

4.3 KB Curation

The goal of this paper is to generate age-appropriate *MCQs* for vocabulary assessments catering to young language learners. Therefore, it is essential to remove the semantic relationships which are 1) inappropriate, 2) rare to observe and 3) inherently noisy from a KB. We handle this problem in a systematic way.

Theme Specific Retrieval and Filtering Based on Children's Book Test: First, we retrieve the part of the KB based on themes relevant to young learners such as fruits, animals, vegetables, transport, etc. Next, from this theme specific KB, we filter the edges where either the source node word or the target node word is part of the Children's Book Test.

Supervised Learning for Filtering Noisy and Irrelevant Links: We use a supervised learning approach to filter out noisy or irrelevant edges. This process begins with crowd-sourcing annotators to manually label the edges as relevant or not. After the manual annotation, we train a binary classifier on the annotated links. The features we pick are, 1) Edge relation, 2) Cosine similarity between source word and target word from word embedding vectors and 3) Weight or confidence score on the edges, if present. After this step, the curated KB is relatively free from noisy and inappropriate data and can be used for *MCQ* generation for YL.

In this section, we first present the method used for automatic seed word selection. Next, we present the strategy used for hard and easy *MCQ* generation.

4.3.1 Seed Words Selection for Each *MCQ* Type:

Question	Correct Options Before Adding Missing Edges	Distractors Before Adding Missing Edges	Nodes Added to Correct Answers	Nodes removed from distractors
Fruit hasA peel?	{lemon, orange, banana, apple}	{melon, lime, pineapple, pumpkin, pear, pomegranate, avocado, plum, ...}	{avocado, melon, lime}	{pumpkin, pomegranate, pear, pineapple}
Tools usedFor cut?	{knife, saw}	{chisel, screw, axe, hoe, ... }	{chisel, axe}	{}
Food hasA crust?	{bread,pie}	{ fruit, mushroom, snack, candy, loaf,......}	{}	{loaf}
Insect capableOf fly?	{butterfly, bee}	{ grasshopper, wasp, bumblebee, tick, worm, ...}	{wasp}	{grasshopper, bumblebee}

Table 1: Examples of Missing Edges Removed

This process involves two steps, 1) Selecting words which are representative of semantic categories such as 'mammal', 'fruit', and 'bird', and 2) Selecting the child nodes of these semantic categories based on a criterion. We employ graph based heuristics to select words corresponding to semantic categories. Words that have a relatively high degree for incoming isA relations, and a relatively low degree for outgoing isA relations qualify as semantic categorical words. Next, we pick the child nodes of these semantic category words which have a relatively high number of edges for $usedFor$, $capableOf$ and $hasA$ relationships. Thus, we generate a list of seed words which we use to create $MCQs$.

4.3.2 Method for Handling Missing Edges:

As described earlier, the curated KB is processed to remove noisy and irrelevant data before MCQ generation. However, YL-KB still contains missing edges. Because of this, some nodes which are correct answers show up as distractors instead. For example, as shown is Figure 1, for question “Which of the following has claws?” correct options are {*bear, dog, ...*} and distractors are {*cow, seal, elephant, bat, ...*}. Due to the missing edge $hasA(seal, claw)$, *seal* becomes a distractor even though it is a correct answer. Our hypothesis for adding missing edges is that if there is a missing edge from words $w1$ to word $w2$ of relation r, then the cosine similarity score between $w1$ and $w2$ must be approximately similar to the cosine similarity score between others words connected to word $w2$ with the same relation r. For adding missing edges, we performed several simulations for the cosine similarity scores (ρ), their means (μ), and their standard deviation (σ) and obtained the following: if ($\rho \geq \mu$) then we assume a valid link; if ($\mu - \sigma \leq \rho < \mu$) we are not sure about the quality of the link; and if ($\rho \leq \mu - \sigma$), we characterize it as an invalid link.

4.3.3 MCQ Generation Method

Our hypothesis for MCQ is that it should have distractors that do not share any common properties with the correct answer. To ensure some confidence in discontinuity between an answer and distractors we leverage the idea of finding non-overlapping graph communities within words in YL-KB. We take the YL-KB graph as a directed graph, ignoring the relationship labels on the edges and use CNM [10] to find communities. For each community, we do a one-hop expansion of each node in that community and remove repeated nodes in this set of expanded and original nodes. Thus, we get new nodes that belong to other communities. We call them leading nodes, as they form a bridge between the communities. To generate $MCQs$, we find the community for each seed word, and its leading nodes. In this way, we can move from a seed word to a related community, if a path between a chosen leading node and a seed word exists. To generate distractors for the seed word and for a seed relation, we pick words from the related communities which are related to other words in their community using the same seed relation.

5. EXPERIMENTS AND EVALUATION

In this section, we present the experiments we conducted for evaluating our proposed approach.

5.1 Experimental Setup

As mentioned earlier, we curated the ConceptNet to create YL-KB. It has age-appropriate themes relevant for young language learners as specified by [11] such as bird, fruit, vegetable, color, insect and animals. We then picked edges labeled with *isA*, *hasA*, *atLocation*, *synonym*, *antonym*, *usedFor*, and *capableOf* relationships. The edges were filtered where either the source node word or target node word was not part of the children's book test [26] for the purpose of filtering inappropriate words.

After the theme specific KB was curated using the corpus [26], we employed a supervised learning technique, specifically a binary multi-layer perceptron implementation from Scikit-learn [25] for filtering of irrelevant and noisy edges. The attributes we used for training the classifier were, 1) source node word, 2) target node word, 3) relationship type, 4) number batch cosine distance [28] and 5) edge weight coming from ConceptNet. Out of total 27070 edges across different relationship types, we picked 28% as the training set, 12% as validation set and rest 60% as test set. For the training set, we asked human annotators to annotate the edges as relevant or irrelevant. The trained classifier had precision and recall for both classes (relevant and irrelevant) around 84% and F-score of around 0.83 on the validation set. After filtering the edges, we were left with about 50% edges that were appropriate, which corresponds to YL-KB. We also added missing edges based on the strategy discussed in Section 4.3.2. For example, we were able to connect nodes $avocado, melon, lime$ to node $peel$ with relation $hasA$ using our strategy. Few other examples are as shown in the Table 1.

From YL-KB, using the methods described in Section 4, we generated correct answers and distractors. For each seed word, we could generate questions in the range of thousands.

5.2 Experiment Design

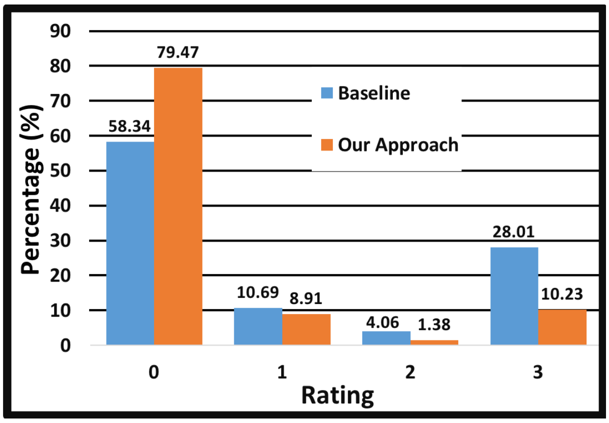


Figure 3: Validation Statistics of Baseline (vanilla ConceptNet) with our approach

In order to establish the efficacy of our approach, we conducted a question usability validation. We also conducted validations which compared our approach with a baseline. For all the validations, we had three volunteers manually annotate the questions. All volunteers were in the age group of 25 − 35 years and had a higher education degree where English was the medium of instruction.

5.2.1 Baseline:

We used vanilla ConceptNet, without applying any filtering to handle noisy or missing edges, to generate *MCQs* using our logic to create correct answer and distractors. We generated 600 questions using our approach (filtered KB i.e, YL-KB) and this baseline approach (without filtered KB) , keeping the same number of questions per word. We asked each annotator to manually annotate all the 600 questions based on usability of the questions on a rating scale of 0 to 3, where 0 corresponds to "no problem with correct answer and distractors", 1 corresponds to "no problem with correct answer and there is a problem with only one distractor", 2 corresponds to "no problem with correct answer and there is a problem with two distractors", and 3 corresponds to "either there is a problem with correct answer or all the distractors".

5.2.2 Question Usability Validation:

The experimental setup and rating score criteria in this validation was the same as described in Baseline. This validation set had 300 questions each from baseline and our approach, i.e. 600 questions in total.

5.3 Results & Discussion

In this section, we report the results of validations we conducted. Figure 3 compares the average annotator percentage for each rating between Baseline (vanilla ConceptNet) and our approach. The difference of 21% in rating 0 and 17% in rating 3 signifies that the *MCQs* generated using vanilla ConceptNet require more revision than *MCQs* generated using our approach due to noisy and missing links. We observe an inter-annotator Fleiss Kappa agreement of 0.56 i.e, a *moderate* inter-annotator agreement. Although this validation was done to compare the usability of generated *MCQs*, however, all the annotators reported that the relatedness of distractors with the correct answer was low in Baseline compared to our approach.

Based on annotation data and interviews conducted with annotators, we infer that some of the ambiguity and less than perfect annotation results arise because of each annotator's individual perspective on word meanings. The observation reiterate why vocabulary assessment, especially for young learners, is a hard problem space, since words are not fixed units of meaning, and can be interpreted differently based on the context they occur in, or on individual perceptions.

6. CONCLUSION

In this paper we presented a system that uses a curated common sense knowledge base for young learners in combination with graph based inferencing to automatically generate *MCQs* for vocabulary assessments. We tested our system extensively by comparing human inter-annotator agreements on a large set of system generated *MCQs*, and observed moderate agreement on the *MCQs*. These initial results are very encouraging to conduct further investigations into how we can build such systems which can generate more complex questions, generate more personalized vocabulary assessments etc. We would also like to look at how this kind of a framework affects the generation of assessments in different modalities (image, audio, video etc.) which are so prevalent in early childhood learning curricula.

7. REFERENCES

[1] M. Al-Yahya. Ontology-based multiple choice question generation. *The Scientific World Journal*, 2014, 2014.

[2] T. Alsubait, B. Parsia, and U. Sattler. Mining ontologies for analogy questions: A similarity-based approach. In *OWLED*, 2012.

[3] T. Alsubait, B. Parsia, and U. Sattler. A similarity-based theory of controlling mcq difficulty. In *e-Learning and e-Technologies in Education (ICEEE), 2013 Second International Conference on*, pages 283–288. IEEE, 2013.

[4] T. Alsubait, B. Parsia, and U. Sattler. Generating multiple choice questions from ontologies: Lessons learnt. In *OWLED*, pages 73–84. Citeseer, 2014.

[5] R. C. Anderson and P. D. Pearson. A schema-theoretic view of basic processes in reading comprehension. *Handbook of reading research*, 1:255–291, 1984.

[6] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, pages 722–735, Berlin, Heidelberg, 2007. Springer-Verlag.

[7] I. L. Beck, M. G. McKeown, and L. Kucan. *Bringing words to life: Robust vocabulary instruction*. Guilford Press, 2013.

[8] R. Boulware-Gooden, S. Carreker, A. Thornhill, and R. Joshi. Instruction of metacognitive strategies enhances reading comprehension and vocabulary achievement of third-grade students. *The Reading Teacher*, 61(1):70–77, 2007.

[9] J. C. Brown, G. A. Frishkoff, and M. Eskenazi. Automatic question generation for vocabulary assessment. In *Proceedings of the conference on*

Human Language Technology and Empirical Methods in Natural Language Processing, pages 819–826. ACL, 2005.

[10] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70:066111, Dec 2004.

[11] EngageNy. *EngageNy*, 2017.

[12] V. EV and P. S. Kumar. Automated generation of assessment tests from domain ontologies. 2016.

[13] C. Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[14] J. L. Fleiss, B. Levin, and M. C. Paik. *Statistical methods for rates and proportions*. John Wiley & Sons, 2013.

[15] D. M. Gates. How to generate cloze questions from definitions: A syntactic approach. In *2011 AAAI Fall Symposium Series*, 2011.

[16] Q. Guo. *Questimator: generating knowledge assessments for arbitrary topics*. PhD thesis, Carnegie Mellon University Pittsburgh, PA, 2016.

[17] A. Hoshino and H. Nakagawa. A real-time multiple-choice question generation for language testing: a preliminary study. In *Proceedings of the second workshop on Building Educational Applications Using NLP*, pages 17–20. ACL, 2005.

[18] Y.-C. Lin, L.-C. Sung, and M. C. Chen. An automatic multiple-choice question generation scheme for english adjective understanding. In *Workshop on Modeling, Management and Generation of Problems/Questions in eLearning, the 15th International Conference on Computers in Education (ICCE 2007)*, pages 137–142, 2007.

[19] H. Liu and P. Singh. Conceptnet &mdash; a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226, Oct. 2004.

[20] R. Mitkov, H. LE AN, and N. Karamanis. A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering*, 12(2):177, 2006.

[21] J. Mostow and H. Jang. Generating diagnostic multiple choice comprehension cloze questions. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 136–146. ACL, 2012.

[22] W. E. Nagy. Vocabulary processes. *Handbook of reading research*, 3(269-284), 2000.

[23] S. Nam, G. Frishkoff, and K. Collins-Thompson. Predicting short-and long-term vocabulary learning via semantic features of partial word knowledge. *Ann Arbor*, 1001:48109.

[24] A. Papasalouros, K. Kanaris, and K. Kotis. Automatic generation of multiple choice questions from domain ontologies. In *e-Learning*, pages 427–434. Citeseer, 2008.

[25] 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. *J. Mach. Learn. Res.*, 12:2825–2830, Nov. 2011.

[26] F. Research. The children's book test corpus from facebook, 2016.

[27] D. Seyler, M. Yahya, and K. Berberich. Knowledge questions from knowledge graphs. *arXiv preprint arXiv:1610.09935*, 2016.

[28] R. Speer, J. Chin, and C. Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. pages 4444–4451, 2017.

[29] S. Supraja, K. Hartman, S. Tatinati, and A. W. Khong. Toward the automatic labeling of course questions for ensuring their alignment with learning outcomes.

[30] E. Vinu and P. S. Kumar. Improving large-scale assessment tests by ontology based approach. In *FLAIRS Conference*, page 457, 2015.

Estimating the Treatment Effect of New Device Deployment on Uruguayan Students' Online Learning Activity

Cecilia Aguerrebere
Fundación Ceibal, Uruguay
caguerrebere@ceibal.edu.uy

Cristóbal Cobo
Fundación Ceibal, Uruguay
ccobo@ceibal.edu.uy

Jacob Whitehill
Worcester Polytechnic Institute, USA
jrwhitehill@wpi.edu

ABSTRACT

When implementing large-scale educational computing initiatives (e.g., One Laptop Per Child) it is vital to allocate resources for training, support, and device deployment judiciously. One question that arises is how learners' engagement with online educational resources is affected by receiving a new computer; do the benefits justify the costs? In this paper, we perform a quasi-experimental analysis to measure the effect of new device deployment on students' online learning activity, operationalized as either the number of interaction events with their LMS, or the number of attempted exercises in their math ITS. The focus is on 6th-grade learners in Uruguay, which to-date has delivered over 750,000 computers to pupils nationwide. Our results suggest that, relative to learners' online learning activity before device deployment, the absolute effects are small but the relative effect are stat. sig. and surprisingly strong: the estimated relative increase on 2016 students' overall LMS activity is 49%. The effects are positive for both 2015 and 2016 and persist several months after device delivery. Moreover, we find that students attempt to solve stat. sig. more (88%) math problems during the month after they receive a new device. We discuss possible reasons and implications for large-scale educational computing programs.

Keywords

One Laptop Per Child; quasi-experimental design

1. INTRODUCTION

During the past 15 years, there have been numerous large-scale educational interventions worldwide – most notably the One Laptop Per Child (OLPC) [16] and One Tablet Per Child (OTPC) [23] programs – that distribute computers to disadvantaged learners to help them bridge the digital divide and achieve better learning outcomes. Early on, such programs were often viewed as a panacea to equalize education worldwide, and indeed some studies have shown that they can boost learners' writing [15] and math [5] skills, verbal fluency [3], basic cognitive processes [3], and self-efficacy [20]. More subtly, they can also help learners to contribute educational content of their own [11] in an educational ecosphere dominated by Western, English-speaking content-makers.

Above all, however, independent evaluations of OLPC and related programs have shown that achieving meaningful learning gains requires more than just giving students laptops and hoping for a positive change [4, 24, 13, 22]. In order for these initiatives to work, it is vital to provide teachers with training on how to make good use of them as part of the curriculum [6]. Computers can break down, and it is important to provide both hardware and software support to ensure these devices remain usable [21]. Finally, even the best maintained device will eventually become obsolete, and thus money for new *device deployment* must be budgeted.

Effectively implementing large-scale educational computing initiatives requires that resources be apportioned judiciously. One question that arises is: **How are learners' interactions with online educational resources affected by receiving a new laptop or tablet computer?** Distributing computers to every student is expensive, and it is important to establish that they are worth the cost. There are several reasons why new devices might impact learners' behavior: (1) *Different affordances*: the new device may offer new features that enable new kinds of interaction. (2) *Novelty*: the mere fact of receiving a shiny new device may incite learners to use it (at least temporarily). (3) *Replacement of broken hardware*: receiving the new device can enable learners simply to *resume* accessing online content.

One way to measure the effect of new device deployment would be to conduct a randomized-controlled trial (RCT), i.e., randomly select a set of students to whom to give a new device at random times throughout the school year, and compare the outcomes of students who received a new device to those who didn't. However, this would be problematic for logistic, political, and ethical reasons, since some people might believe *a priori* that the benefits of receiving a new device could be significant. In this paper, we instead pursue a quasi-experimental approach: One of the potential opportunities offered by educational data-mining is to estimate the causal impact of different interventions from *observational* datasets, i.e., data that were collected containing many covariates/features but *without* random assignment of treatments to participants. Over the past few decades, a variety of techniques have been developed for this purpose, including propensity score matching [18], principal stratification [9], regression discontinuity analysis [12], and others [19]. Such methods are only applicable in specific contexts, such as in a *natural experiment* in which an exogenous event causes the treatment assignment to be *essentially* random w.r.t. any variable that could conceivably influence the outcome of the treatment itself (i.e., potential confounds). In such situations, random assignment can be imputed *post hoc*, and treatment effects can be estimated by comparing the treated subjects to the untreated ones.

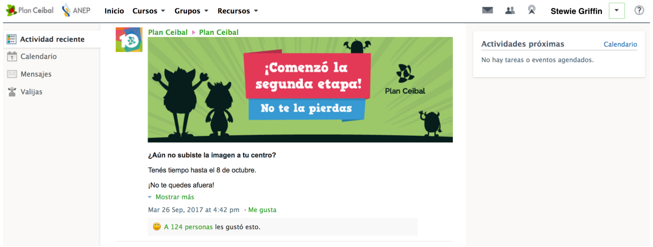
Our paper represents a **case study in quasi-experimental educational data-mining**: We examine how learners, who received computers as part of OLPC, are affected by new device deployment in terms of their interactions with online educational content. Our geographical focus is on Uruguay, which was one of the largest (in terms of number of pupils receiving a laptop) participants in the OLPC program [13]. During 2007-2016, the government of Uruguay together with the Plan Ceibal organization distributed laptops and tablets to over 750,000 pupils nationwide. The nearly universal implementation of this program within Uruguay offers an opportunity to estimate a "new device effect" since there is no selection bias of who receives a new device. We assess the impact of new device deployment on two dependent variables: (1) the *total number of interaction events* with their learning management system (LMS); and (2) the *number of attempted math exercises* within their mathematics intelligent tutoring system (ITS); in prior research, the number of attempted exercises in ITS has been shown to correlate with students' performance on standardized math tests [7, 19, 8].

1.1 Related work

Many studies have examined the educational impact of OLPC programs in general; however, the issue of *new device deployment* within educational computing initiatives and how they are perceived by and affect users, has received much less attention. Oliver & Goerke [17] conducted a survey of engineering and business students in Australia, Ethiopia and Malaysia to assess learners' willingness to adopt a new device (the HP iPAQ) for educational purposes. One notable result was that female students in the participating countries indicated lower willingness to trial the new devices than their male counterparts. In addition, Lai, et al. [14] surveyed students in Hong Kong on their willingness to adopt new educational technology and found that device compatibility with the students' perceived learning styles would affect their likelihood of using it. Neither study examined quantitatively how new devices impact learning behaviors. Hence, these works can be seen as complementary to ours in that they seek to describe the *interactions* between different types of learners and different types of educational technology that might jointly influence their impact on learning.

1.2 OLPC in Uruguay & Plan Ceibal

Since 2007, Plan Ceibal has provided a computer to almost every student in primary and secondary schools in Uruguay, and also ensured Internet access in schools and as well as public access-points. The initial goals were to reduce the digital divide, promote digital inclusion, and ensure the integration of ICT in education. Since 2011, Plan Ceibal has focused on providing the educational community with a wide range of digital tools, such as an LMS, an ITS for mathematics, a digital library, a videoconference system to teach English as a second language and facilitate collaboration, etc. The LMS managed by Plan Ceibal is called "CREA2" and is shown in Figure 1. The math ITS is called "PAM".



¿Aún no subiste la imagen a tu centro?

Tenés tiempo hasta el 8 de octubre.

¡No te quedes afuera!

Mar 26 Sep, 2017 at 4:42 pm

A 124 personas les gustó esto.

No hay tareas o eventos agendados.

Figure 1: The CREA2 LMS used by Plan Ceibal. Students can submit homework, send messages to teachers and other students, view content posted by their teachers, etc.

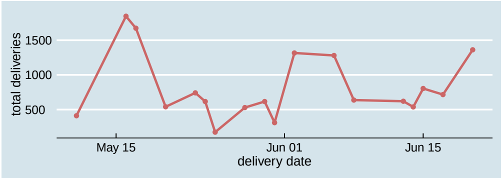


Figure 2: # devices delivered for 6th grade students in 2016

1.3 Device Delivery Process

Students' devices are upgraded several times during the 9 years of basic education (ages 6-14 years): First graders (6 years old) receive a tablet which they use for two years. In 3rd grade they receive a new tablet which they use for one year only. In 4th grade the tablet is replaced by a laptop, which students use for three years. The laptops are then replaced during either 6th or 7th grade (see Figure 2). Within each school, most (90% of primary and 70% of secondary) students in each classroom receive their new devices at the same time. The schedule is set by Plan Ceibal; larger schools have priority, along with schools located close to the delivery path, etc. While the delivery process is not strictly random, *the delivery dates are independent of many factors* including students' prior LMS and ITS activity, the curriculum the children are learning, dates of examinations, holidays, life-changing events for students, etc. This helps to remove many potential confounds that would impede the inference of treatment effects.

2. EXPERIMENTAL ANALYSIS

We investigate the effect of new device deployment in terms of two dependent variables: (1) Δ **LMS Interaction Events**: The increase in students' activity (total number of interaction events) with the CREA2 LMS *after* receiving their new device compared to their activity *before* receiving it. (2) Δ **ITS Attempted Exercises**: The increase in the number of math exercises that students attempt to solve with the PAM ITS *after* versus *before* receiving their new device.

2.1 Dataset

Δ **LMS Interaction Events**: The dataset includes each student's activity in the CREA2 platform on each day of

Year	Learners	Active L.	Deliv. dates	Sch.	Classr.
2015	13329	7276	21	526	809
2016	25898	16962	18	810	1378

Table 1: Plan Ceibal dataset: total # of considered learners, *active* learners, delivery dates, schools, and classrooms containing students who received devices that year.

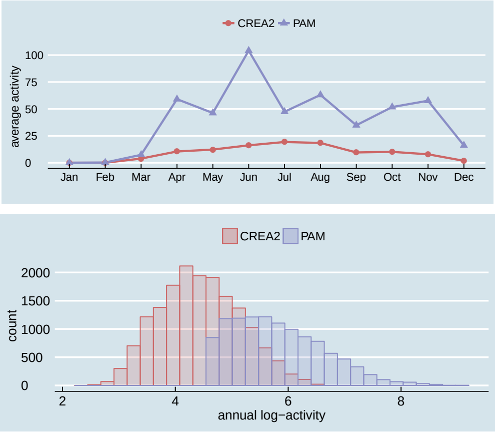


Figure 3: Activity levels in the CREA2 (# actions) and PAM (# attempted exercises) platforms during 2016. **Top:** Average per-student monthly activity. **Bottom:** Histogram of the logarithm of the total annual activity per student.

2015 and 2016, as well as the delivery dates of new devices during that period. A large fraction of the students almost never used the platform. Our focus is on the impact of new devices on *active* students; hence, we limit the universe of study to students who accessed the platform on at least 10 different days in a given year (this is the *active user* definition used at Plan Ceibal). We note that, even with this constraint, the median CREA2 activity level per month is low: only 7 total actions. In addition, we focus exclusively on 6th graders (11 years old), who are the most active CREA2 users. Finally, we only consider delivery dates on which at least 5 new devices were delivered. Table 1 summarizes the sample sizes considered for each dataset for 6th grade.

Δ **ITS Attempted Exercises**: Data were available for 2016 (but not 2015) on 6th-grade students' total math exercises attempted each day. The universe of study is limited to those students who attempted at least 100 exercises in the year (*active user* definition at Plan Ceibal). In addition, during 2017 (but not 2016), the numbers of *correct* and *incorrect* attempted exercises are also available. Figure 3 shows the overall activity levels in CREA2 and PAM. The platforms are offered as a recommended tool for teachers, but their use is not mandatory. Plan Ceibal provides tutorials promoting their use, which are independent of device delivery dates.

Δ **CREA2 activity: 1 month after v. 1 month before**: As a preview of our more detailed analyses below, Figure 4 shows students' *delta* behavior, i.e., their CREA2 activity during the month *after* each delivery date t, minus their activity during the month *before* t. The blue curve shows the deltas for learners who *received* a device on t (along with the number of such students), and the red curve shows students who received a device (at least one month) *after* t. If giving students a new device has a positive impact on CREA2 activity levels, then we expect the blue curve to be higher than the red curve (which it is).

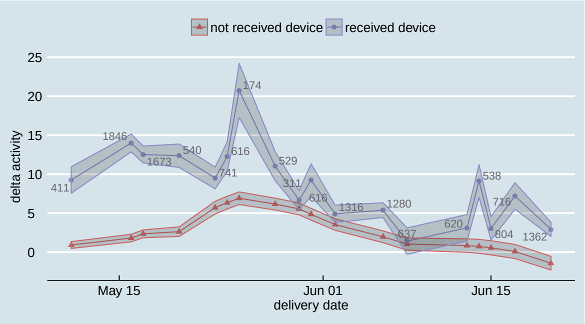


Figure 4: Increase between the average monthly activity in CREA2 after and before each delivery date t, for students who received (blue) and did not receive (red) a device on t, during 2016. The outer bands correspond to 2 standard errors for each mean estimate. The numbers listed by each blue point report how many students received a device on t.

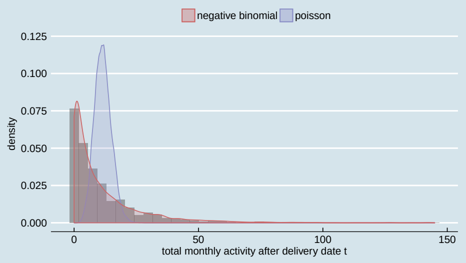


Figure 5: Normalized histogram of the monthly activity after t (for delivery date 2016-05-11).

2.2 Methodology

This is a quasi-experimental study enabled by the *delayed treatment* design [10] that was used in deploying new devices to students: Almost every student in every school who participates in Plan Ceibal eventually receives a new device; hence, there is no selection bias as to who enrolled in the program. In particular, (almost all) students within the same grade of the same school receive their devices on the same date, but these dates are essentially random *across* schools. In particular, the delivery dates are independent of the classroom curriculum and students' prior activity on the CREA2 and PAM platforms. In our analysis, we thus study the effect of device delivery at each delivery date separately and then average these estimates to estimate the average treatment effect across all dates. We do note, however, that our analysis is not immune to all possible confounds, e.g., a relationship between the date of device delivery and whether the school is located in an urban or rural environment.

2.2.1 *Data model*

Each learner's activity in the CREA2 and PAM platforms consists of *count* data. Suitable models for counts include the Poisson and the negative binomial distributions. The advantage of the negative binomial is that the variance of the distribution can be set independently of its mean to account for overdispersion of the data. Figure 5 shows the normalized histogram of the total CREA2 activity after a given delivery date t, overlaid onto Poisson and negative binomial probability density functions fit to the histograms using maximum likelihood estimation (Poisson log-likelihood = -70647.08, negative binomial log-likelihood = -22066.75). This comparison shows the clear overdispersion of the considered data, which makes the negative binomial a more accurate approximation than the Poisson model.

Multi-level modeling: Because for each delivery date we are considering students in the same school, and possibly in the same classroom, the activity data for them will be correlated. Hence, a multi-level modeling approach is employed where the classroom effect on the student's activity, often determined by the teacher, will be modeled as a random effect. We only consider deviations of the *intercept* of a classroom from the overall intercept; random *slopes* are not considered. Therefore, we propose to model student i's activity N months after the delivery date t (i.e., between $t + ((N - 1)$ months) and $t + (N$ months)) as a negative-binomial random variable with expected value A_{it} given by:

$$\log(A_{it}) = e_t + \gamma_{0t} b_{it} + \gamma_{1t} d_{it} + C_t, \qquad (1)$$

(capital letters denote random variables and lower-case denote fixed values). The case $N = 1$ corresponds to the activity during the month right after the delivery date. We define a "month" to be 4 weeks (28 days). e_t is the baseline activity in the same time period considered for A_{it}. b_{it} is student i's activity during the month before the delivery date t. d_i is a boolean variable taking value 1 if student i got a new device on t and 0 otherwise. The fixed effects γ_{0t} and γ_{1t} represent the effect on the activity N months after t, of the activity during the month before t, and the device delivery, respectively. The random effect of classrooms is represented by the random variable C_t, assumed to follow a zero-mean Gaussian distribution and standard deviation σ_{C_t}. A nested classroom-school random effect was also explored, but it was discarded because the results were very close.

Gender: Oliver & Goerke [17] found that female students (in Australia, Ethiopia and Malaysia) reported different attitudes towards educational technology than their male counterparts. Might device deployment affect Uruguayan girls and boys differently in terms of CREA2 activity? To investigate, we extended the Model 1 with a boolean variable g_{it} representing the student's gender as well as an interaction between g_{it} and the device delivery variable d_{it}.

Treatment effect: When computing the device delivery effect of a given delivery date t, we compare students who received a device on t (treatment group), to students who received a device on $t^* > t+(N$ months) (control group). In particular, we make sure to exclude from the control group those students whose treatment occurred within N months of students in the treatment group. This analysis is consistent with a delayed treatment design.

2.2.2 Combining per-date estimates

For each of the M considered delivery dates, we compute the maximum likelihood estimator (MLE) of the device effect γ_{1t}, as well as its associated standard error $SE_{\gamma_{1t}}$. Because different number of students receive/do not receive their devices on each date, the standard errors $SE_{\gamma_{1t}}$ will vary across dates. We model the M estimates $\{\gamma_{1t}\}_{t=1,\ldots,M}$, as independent samples of Gaussian random variables with equal mean γ_1 and different standard deviations $SE_{\gamma_{1t}}$. Then, the MLE of the device delivery effect $\hat{\gamma}_1$ is given by averaging the individual γ_{1t}'s weighted by the inverse square of their standard errors. From $\hat{\gamma}_1$ and its standard deviation we can compute confidence intervals, and perform a t-test to assess the statistical significance of the device delivery effect [2].

To ensure that the treatment effect estimates $\{\gamma_{1t}\}_{t=1,\ldots,M}$ across delivery dates are statistically independent, each group of students belonging to the same classroom is used to estimate the treatment effect for one delivery date only. That is, all the classrooms under consideration are partitioned over delivery dates, and the treatment effect for each date is computed only from the students assigned to that date. Some of these students will be in the treatment group (those who received the device that day) and others will be in the control group (those who received the device later).

To partition students across delivery dates, we used a greedy algorithm whereby one classroom is assigned to a delivery date at a time: For each classroom, one of the M delivery dates t is chosen with probability p_t, which is inversely proportional to the total number of students already assigned plus the total potential number of students that could be assigned to each date – thus favoring dates not yet assigned and with few potential students. We ran this procedure 100 times and selected the assignment with smallest variance in the number of students assigned per date, which helps to avoid possible numerical issues in model estimation.

Implementation: Models were fit using the *glmer.nb* function of the R *lme4* package. To detect possible convergence problems, each experiment was run using several different optimizers and consistent results were verified [1].

3. RESULTS I: LMS INTERACTIONS

Table 2 shows the estimated effects (averaged over all delivery dates) of delivering a new device on the number of CREA2 interaction events, for 2015 and 2016. Since the computed effects are in *logarithmic* scale (see Equation 1), a log-effect of 0 corresponds to exp(0.0) = 1.0 in the original scale, i.e., no impact on CREA2 activity, whereas 0.76 equals exp(0.76) = 2.1 in the original scale, i.e., a 110% activity increase. Table 3 shows the average, over all delivery dates, of the rest of Model 1 parameters in original scale.

Even after accounting for class-specific random effects as well as students' prior baseline activity levels, **we observe a clear CREA2 activity boost in the 1 month following the device delivery** (first row in Table 2): a relative effect of exp(0.76) = 2.14 (114% increase) for 2015 and exp(0.40) = 1.49 (49% increase) for 2016. In other words, while the absolute increases are low (due to the low overall CREA2 activity usage – Figure 3), the relative effect is high. Results for 2016 tend to be less noisy because the activity levels are larger compared to 2015. Though present in both years, the effect clearly decreases from 2015 to 2016.

	2015						2016					
	all class.			class.≥10 stud.			all class.			class.≥10 stud.		
N-months	$\hat{\gamma}_1$	$SE_{\hat{\gamma}_1}$		$\hat{\gamma}_1$	$SE_{\hat{\gamma}_1}$		$\hat{\gamma}_1$	$SE_{\hat{\gamma}_1}$		$\hat{\gamma}_1$	$SE_{\hat{\gamma}_1}$	
1	0.76	0.12	***	0.69	0.14	***	0.40	0.04	***	0.41	0.04	***
2	0.84	0.13	**	0.80	0.15	*	0.20	0.04	**	0.19	0.04	*
3	0.83	0.27		0.50	0.29		0.31	0.14		0.28	0.13	

Table 2: Effects of delivering a new device to each student on their CREA2 activity for academic years 2015 and 2016, over a period of $N = 1, 2, 3$ months, in logarithmic scale. Weighted averages, together with their corresponding standard errors, are reported. Significance codes: 0 (***), 0.001 (**), 0.01 (*), 0.05 (.), 0.1, () 1.

	Fixed eff.		Random eff.		Fixed eff.		Random eff.
year	e	γ_0	σ_C	**year**	e	γ_0	σ_C
2015	1.17	1.06	3.18	**2016**	5.92	1.02	2.88

Table 3: Average of the model parameter estimates among delivery dates in the original scale.

Temporal evolution: The second and third rows of Table 2 show the estimates of the effect of device delivery in students' monthly CREA2 activity 2 and 3 months after the delivery date, respectively. The effect is still present two months after the delivery date, and it appears to be stable in 2015 and to decrease in 2016. The effect is not statistically significant three months after delivery; this may due to the small number of samples available at that time. (Note that examining $N > 3$ was not possible since there were too few students who had not yet received a device who could serve as a control group.)

Classroom size: No significant differences are observed when comparing the estimates considering all classrooms or only those with at least 10 students (see Table 2).

Highly active students: We also conducted the analysis on only those students who accessed CREA2 on at least 25 different days in 2016. (Note that 2015 data could not be analyzed due to small sample size.) The results were consistent with what we found above for all students, with 43% activity increase right after receiving the new device.

Activity change: To investigate what *kinds* of CREA2 activities were affected, Figure 6 shows the percentage of the monthly activity increase, at different time points (the month before t, the month after t, and two and three months after t), of the students who received the device on t relative to those who received it after t. For instance, the average increase in the number of comments posted during the month following t, by the students who received their new device on that date, was 70% larger than that of students who received their device later in the year. For some activity types, the boost is larger and remains longer in time (e.g., comments posted, item submissions). Note that the sum over all activities at the first time-point ($t - 1$) should be close to zero, denoting similar total activity for all users before device delivery.

Gender effect: Using the extended model to support possible gender effects, we found a clear difference on the total activity: girls performed about 32% more total CREA2 activity compared to boys in 2015 (and 28% more in 2016). We observed no significant difference, however, in the effect of device delivery between boys and girls.

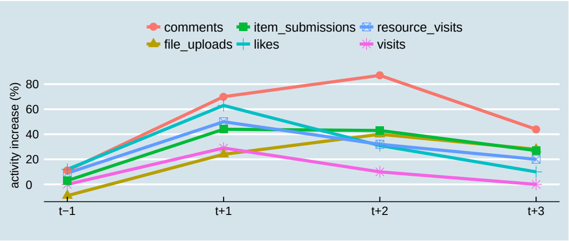


Figure 6: Monthly activity increase (%), at different time points, of students who received the device on t relative to those who did not receive it.

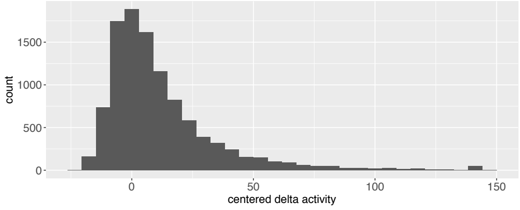


Figure 7: Histogram of Δ CREA2 activities of treated students (over all delivery dates) w.r.t. median activity of untreated students.

Who drives the effect?: Was the the boost in CREA2 activity driven by a large increase among a small number students? To explore this question, we calculated the change (Δ_i) in CREA2 activity level for each student i before/after treatment, minus the median change in activity level for all untreated students. We then computed a histogram over the Δ_i values over all students and delivery dates in 2016. The histogram is shown in Figure 7. Since there is a positive and statistically significant treatment effect, the mean of the histogram is greater than 0. The histogram also shows smooth gradation from small effects to large effects and provides evidence that the average treatment effect is due to increased activity levels among many students, not just a few.

4. RESULTS II: MATH PROBLEM ATTEMPTS

Similar to the analysis of device deployment on students' CREA2 LMS activity, we used the same model (Eq. 1) to assess the potential impact on the number of *attempted exercises* in the PAM math ITS provided by Plan Ceibal.

Results: A positive ($\hat{\gamma}_1 = 0.63$, $SE_{\hat{\gamma}_1} = 0.14$) and statistically significant ($p < 0.01$) effect is observed during the one month following the device delivery, with a 87% increase in the total attempted exercises ($\exp(0.63) = 1.87$). The effects observed two and three months later are small and not statistically significant, suggesting that the boost disappears over time.

5. CORRECTNESS OF MATH EXERCISES

In addition to the number of *attempted* math exercises, we also explored whether receiving a new device helps students to complete exercises *correctly.* Possible reasons include: (1) the new device has a better user interface that helps students avoid careless data-entry errors; and (2) the learners' improved user experience encourages them to practice more often and thereby improve their math skills.

To assess the impact of device deployment on correctness ($0 - 1$ scale) of submitted exercises, it was not possible to use the same methodology as for LMS activity. The reason is that PAM data on correct/incorrect exercises are available only for 2017, and during this year only a small number of 6th grade students received a device. Hence, we resorted to a correlational analysis in which we estimated the change in exercise correctness *without* a control group. In particular, we estimated the treatment effect based on the average accuracy N months after device delivery minus the average accuracy N months before delivery, and averaged across all treatment dates. Because no control group is used (unlike in the previous analyses), there may be confounding factors affecting this analysis.

Results: None of the average delta accuracies for 4th, 5th and 6th grade, computed either on individual students (Δacc_S) or on classrooms (Δacc_C), was statistically significant.

6. DISCUSSION

The results suggest that receiving new devices resulted in a strong relative **increase in learners' CREA2 LMS activity**: 114% in 2015 and 49% in 2016. Within sensitivity analyses based on academic year (2015 and 2016), classroom size, and students' baseline activity levels, we found that the trends were similar: new devices result in increased LMS activity. Moreover, the boost in activity persists up to 3 months after device delivery. We note again that the *absolute* average CREA2 activity levels were very low; hence, the increase may only amount to a few extra logged events (about 10 extra actions per student per month).

Receiving new devices not only increases the activity but also **alters the kind of activities** performed in the platform (Section 3). The fact that device delivery increases (w.r.t. learners who did not receive a device) the number of *comments* and *resource_visits* even more than just *visits* (which reflects merely accessing the CREA2 web page) suggests that learners are engaging more *substantively* with the LMS after receiving their new device.

Within the math ITS, we observed that, during the 1 month after receiving a new device, **learners attempted to solve more (88%) math problems**. However, the results were not statistically significant two and three months after delivery, suggesting that the impact is short-lived. We found no evidence (given the limited data available in 2017) that new devices resulted in higher *accuracy* of attempted exercises.

6.1 Possible explanations

Novelty: Receiving a brand-new device could potentially increase students' motivation to use them, but the effect might diminish over time. In our data, we do observe that the LMS activity boost, as well as the boost in number of attempted ITS math exercises, declines over time (though more strongly for the ITS than for the LMS) after receiving a device declines – which suggests possible novelty effects.

Availability: Oftentimes, devices are not available to students because of recurrent failures. Hence, receiving a new device not only means having a new, more performant one but having a working device at all. It is possible that a student who suddenly (due to device deployment) regains access to a working computer might resume CREA2 activity at a much higher level after receiving it. The strong activity gains we observe are compatible with this hypothesis (though they cannot directly confirm it).

7. SUMMARY AND CONCLUSIONS

We conducted a quasi-experimental analysis (on 24,000 learners over 2 years) to estimate the treatment effect of giving OLPC students new computers. We harnessed the facts that (1) all students were eventually treated, so that there was no selection bias, and (2) the device deployment schedule was random w.r.t. a variety of potential confounds (e.g., students' prior LMS/ITS activity). The main results include:

(1) When students receive a new device, they interact more with their schools' LMS and engage more (attempt more exercises) with their math ITS, compared to learners who had not yet received a device upgrade. To the extent that increased engagement with educational content and practice in solving exercises contributes to students' learning [7, 19, 8], OLPC programs should try to provide students with up-to-date devices in a timely and cost-effective manner.

(2) While conducting these analyses we discovered that the *absolute* baseline activity levels of many learners in the examined dataset were very small. This raises the question of whether teachers are receiving proper training on how to use online learning resources effectively and how to instruct and encourage their learners to engage with them.

(3) Our study indirectly raised the question of how often a new device delivery simply replaces a device that had *broken.* For researchers who wish to assess the potential benefits of OLPC programs, it is important to take into account how many students truly have access to a working device (not just a broken one). For administrators, it underlines how technical support may play an important role in ensuring the success of large-scale educational computing initiatives.

(4) The fact that new device deployment increases CREA2 and PAM activities – even if the effects are transient – is evidence that learners' activities *can* be incited to engage more with educational platforms. One way is through renewed hardware, as explored in this paper. Another way is to help teachers to *use* these platforms more effectively [22].

Future research: It would be interesting to explore whether novelty, new features, or replacing broken hardware contributes more to the overall treatment effect; to this end, it would be useful to ask learners themselves about how they perceive and interact differently with new devices.

8. REFERENCES

[1] Bates, D., Mächler, M., Bolker, B., and Walker, S. Fitting linear mixed-effects models using

lme4. *J Stat Softw 67*, 1 (2015), 1–48.

[2] Borenstein, M., Hedges, L. V., Higgins, J. P., and Rothstein, H. R. A basic introduction to fixed-effect and random-effects models for meta-analysis. *Research Synthesis Methods 1*, 2 (2010), 97–111.

[3] Cristia, J., Ibarraran, P., Cueto, S., A., S., and Severin, E. Technology and child development: Evidence from the one laptop per child program. *IZA Paper no. 6401* (2012).

[4] De Melo, G., Machado, A., and Miranda, A. The impact of a one laptop per child program on learning: Evidence from uruguay. *IZA Paper no. 8489* (2014).

[5] Díaz, A., Nussbaum, M., and Varela, I. Orchestrating the xo computer with digital and conventional resources to teach mathematics. *J Comput Assist Lear 31*, 3 (2015), 202–219.

[6] Fajebe, A. A., Best, M. L., and Smyth, T. N. Is the one laptop per child enough? viewpoints from classroom teachers in rwanda. *Information Technologies & International Development 9*, 3 (2013), pp–29.

[7] Feng, M., Heffernan, N., and Koedinger, K. Addressing the assessment challenge with an online system that tutors as it assesses. *User Modeling and User-Adapted Interaction 19*, 3 (2009), 243–266.

[8] Feng, M., and Roschelle, J. Predicting students' standardized test scores using online homework. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale* (2016), ACM, pp. 213–216.

[9] Gallop, R., Small, D. S., Lin, J. Y., Elliott, M. R., Joffe, M., and Ten Have, T. R. Mediation analysis with principal stratification. *Statistics in medicine 28*, 7 (2009), 1108–1130.

[10] Heath, L., Kendzierski, D., and Borgida, E. Evaluation of social programs: A multimethodological approach combining a delayed treatment true experiment and multiple time series. *Evaluation Review 6*, 2 (1982), 233–246.

[11] Hourcade, J. P., Beitler, D., Cormenzana, F., and Flores, P. Early olpc experiences in a rural uruguayan school. In *CHI'08 extended abstracts on Human factors in computing systems* (2008), ACM, pp. 2503–2512.

[12] Imbens, G. W., and Lemieux, T. Regression discontinuity designs: A guide to practice. *Journal of econometrics 142*, 2 (2008), 615–635.

[13] Kraemer, K. L., Dedrick, J., and Sharma, P. One laptop per child: vision vs. reality. *Communications of the ACM 52*, 6 (2009), 66–73.

[14] Lai, C., Wang, Q., and Lei, J. What factors predict undergraduate students' use of technology for learning? a case from hong kong. *Computers & Education 59*, 2 (2012), 569–579.

[15] Lowther, D. L., Ross, S. M., and Morrison, G. M. When each one has one: The influences on teaching strategies and student achievement of using laptops in the classroom. *Educ. Tech. Research and Development 51*, 3 (2003), 23–44.

[16] Negroponte, N., Bender, W., Battro, A., and Cavallo, D. One laptop per child. In *National Educ. Computing Conference in San Diego, CA. Retrieved April* (2006), vol. 5, p. 2007.

[17] Oliver, B., and Goerke, V. Undergraduate studentsâĂŹ adoption of handheld devices and web 2.0 applications to supplement formal learning experiences: Case studies in australia, ethiopia and malaysia. *International Journal of Education and Development using ICT 4*, 3 (2008).

[18] Rosenbaum, P. R., and Rubin, D. B. The central role of the propensity score in observational studies for causal effects. *Biometrika 70*, 1 (1983), 41–55.

[19] Sales, A. C., and Pane, J. F. Exploring causal mechanisms in a randomized effectiveness trial of the cognitive tutor. In *Proceedings of the 8th International Conference on Educational Data Mining* (2015).

[20] Shank, D. B., and Cotten, S. R. Does technology empower urban youth? the relationship of technology use to self-efficacy. *Computers & Education 70* (2014), 184–193.

[21] Simon, S. Laptops may change the way rural peru learns. *Weekend Edition* (2008).

[22] Therias, E., Bird, J., and Marshall, P. Más tecnologia, más cambio?: Investigating an educational technology project in rural peru. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (2015), pp. 447–456.

[23] Viriyapong, R., and Harfield, A. Facing the challenges of the one-tablet-per-child policy in thai primary school education. *Education 4*, 9 (2013), 23–32.

[24] Warschauer, M., Zheng, B., Niiya, M., Cotten, S., and Farkas, G. Balancing the one-to-one equation: Equity and access in three laptop programs. *Equity & Excellence in Education 47*, 1 (2014), 46–62.

ELBA: Exceptional Learning Behavior Analysis

Xin Du
Eindhoven University of Technology
x.du@tue.nl

Wouter Duivesteijn
Eindhoven University of Technology
w.duivesteijn@tue.nl

Martijn Klabbers
Eindhoven University of Technology
m.d.klabbers@tue.nl

Mykola Pechenizkiy
Eindhoven University of Technology
m.pechenizkiy@tue.nl

ABSTRACT

Behavioral records collected through course assessments, peer assignments, and programming assignments in Massive Open Online Courses (MOOCs) provide multiple views about a student's study style. Study behavior is correlated with whether or not the student can get a certificate or drop out from a course. It is of predominant importance to identify the particular behavioral patterns and establish an accurate predictive model for the learning results, so that tutors can give well-focused assistance and guidance on specific students. However, the behavioral records of individuals are usually very sparse; behavioral records between individuals are inconsistent in time and skewed in contents. These remain big challenges for the state-of-the-art methods. In this paper, we engage the concept of **subgroup** as a trade-off to overcome the sparsity of individual behavioral records and inconsistency between individuals. We employ the framework of **Exceptional Model Mining (EMM)** to discover exceptional student behavior. Various model classes of EMM are applied on dropout rate analysis, correlation analysis between length of learning behavior sequence and course grades, and passing state prediction analysis. Qualitative and quantitative experimental results on real MOOCs datasets show that our method can discover significantly interesting learning behavioral patterns of students.

Keywords

Exceptional Model Mining, MOOCs, Learning Analytics

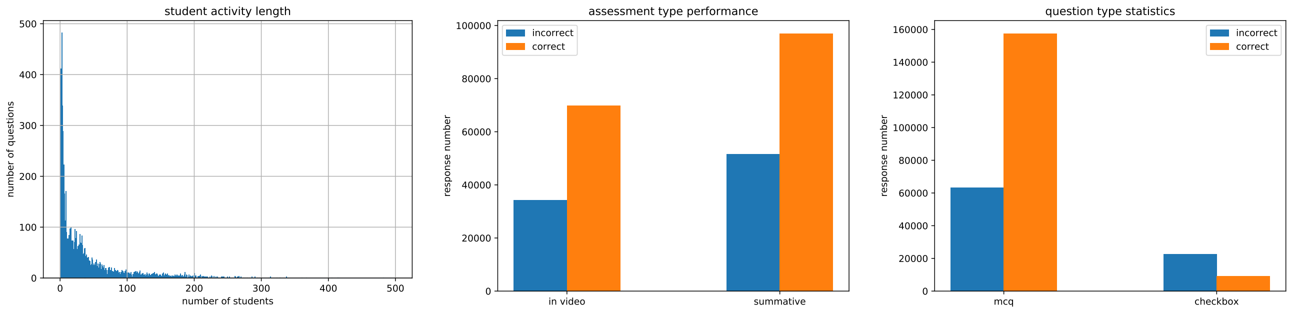
1. INTRODUCTION

Massive Open Online Courses (MOOCs) make it possible for educators to analyze learning behavior of students in multiple views. In contrast to traditional classes, which only have limited learning behavioral records, MOOC platforms such as Coursera, edX and Udacity provide huge amounts of learning behavioral records. These platforms collect very detailed course information and students' learning behavior such as course assessments, peer assignments, programming assignments, forum discussions and feedback [19], which can reflect the knowledge and skill achievements and the study performance of students. Modeling students' learning behavior and trying to discover interesting behavioral patterns are non-trivial. Most recent research is focused on how to predict the learning results based on the learning behavior model. It can help the tutors to design the courses and give specific guidance and assistance to specific students. However, due to the complexity of the behavioral records, there are still several challenges to be overcome:

Individual sparsity. Even when many students are enrolled in a course, the duration of their involvement varies substantially. Figure 1a displays a histogram of assessment question frequencies, which shows an obvious Power-Law distribution [2]. Only a few students participate in hundreds of assessment questions. Most of the students have activity length less than 20 records, which is very sparse. This makes evolutionary activity sequence based user modeling methods [16, 17] ineffective.

Activity inconsistency. Beyond the distribution in activity length of assessment questions, students' learning behavior in forum discussion, click stream and peer review are also shown to follow a Power-Law distribution. In Table 4, we can see that among the 18 courses on Coursera, enrolled students, grades and students who passed the course are highly diverse. This inconsistency makes the data very imbalanced, which results in difficulties for Matrix factorization based modeling methods [24]. These methods might merge infrequent behavior with common behavior.

Content heterogeneity. Behavior diversity is not only shown in activity length and course status, but also shown in informative contents. There are 7 types of assessments and 12 types of questions in the courses, such as video, summative, checkbox and multiple checkbox. Proportions of these assessments and questions are skewed in different courses. On the other hand, students also have varying participation records on these contents. In Figure 2, it is shown that distributions of students are obviously different in specific demographic categories. It is a big challenge for modeling methods to handle these heterogeneous contents for tasks like dropout prediction or passing state prediction.

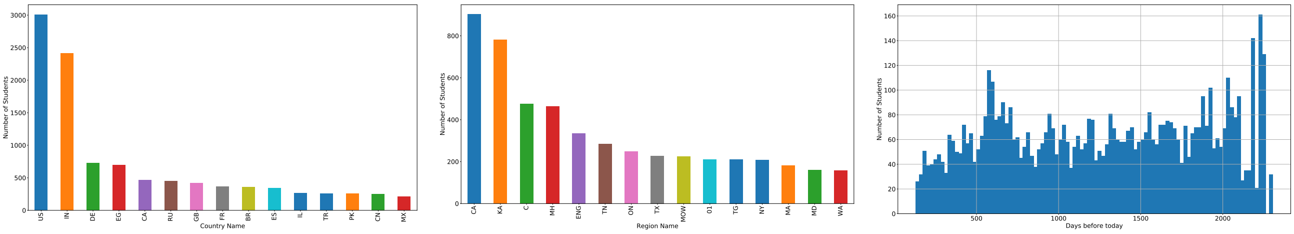


(a) Histogram of the number of assessment questions in which students participate.

(b) Performance per assessment type.

(c) Performance per question type. 'mcq' represents multiple checkbox questions.

Figure 1: Heterogeneity and inconsistency of student behavior.



(a) Number of students by country.

(b) Number of students by regions.

(c) Number of students by join date.

Figure 2: Student distributions across various demographic categories.

To overcome these challenges, we propose to employ Exceptional Model Mining (EMM) [4] for exceptional learning behavior analysis. Instead of looking for anomalies or outliers of individuals, we look for exceptional behavior on the subgroup level [7], which can provide interpretable descriptions such as 'Students: Country = US, Region = Manhattan, Join dates > 365 (days)' having exceptional learning behaviors that are predominantly different from those in the whole dataset. We employ EMM to discover interesting learning behavioral patterns in subgroups. We establish various model classes for specific learning behaviors, such as discovering correlation between length of behavior sequence and course grades, finding out subgroups with exceptional dropout ratio, and looking for specific subsets where the classifier does not perform well. Experimental results on a real dataset illustrate the type of meaningful learning behavioral patterns EMM can discover in MOOCs. This can help us build an improved behavior model in the future research. In summary, our main contributions are:

1. We employ Exceptional Model Mining (EMM) to learning behavior analysis in MOOCs, which can help us to overcome the sparsity, inconsistency and heterogeneity in the behavioral records.
2. We employ several EMM model classes for different tasks to discover exceptional learning behaviors on the subgroup level. Our results show very interesting learning behavioral patterns, which can help the tutors conduct specific guidance and assistance to the students.

2. RELATED WORK

Local Pattern Mining (LPM) [6, 14] is a subfield of data mining, concerned with discovering subsets of the dataset at hand where something interesting is going on. Typically, a restriction is imposed on what kind of subsets we are interested in: only those subsets that can be formulated within a predefined *description language* are allowed. A canonical choice for this language is conjunctions of conditions on attributes of the dataset. Hence, if the records in our dataset concern people, then LPM finds results of the form:

$$\text{Age} \geq 45 \wedge \text{Smoker} = \text{yes} \quad \rightsquigarrow \quad \text{interesting}$$

This ensures that the results we find with an LPM method are relatively easy to interpret for a domain expert: the subsets will be expressed in terms of quantities with which the expert is familiar. We call a subset that can be expressed in such a way a *subgroup*.

Different LPM methods give a different answer to the question what exactly constitutes "where something interesting is going on". The most famous form of LPM is *Frequent Itemset Mining* (FIM) [1], where interestingness is equivalent to occurring unusually frequently: things that happen often are interesting. Hence, FIM finds results of the form:

$$\text{Age} \geq 45 \wedge \text{Smoker} = \text{yes} \quad \rightsquigarrow \quad (\text{high frequency})$$

The methods we are mainly concerned with in this paper, however, seek a more complex concept on the right-hand side of this arrow. The task of *Subgroup Discovery* (SD) [9, 23, 7] typically singles out one binary attribute of the dataset as the *target*: subgroups are deemed interesting if this one target has an unusual distribution, as compared to its distribution on the entire dataset. In our example, if the target column describes whether the person develops lung cancer or not, SD finds results of the form:

$$\text{Smoker} = \text{yes} \rightsquigarrow \text{lung cancer} = \text{yes}$$
$$\text{Age} \leq 25 \rightsquigarrow \text{lung cancer} = \text{no}$$

These subgroups make intuitive sense in terms of our knowledge of the domain. Smokers have a higher-than-usual incidence of lung cancer. People under the age of 25 often have not yet had the chance to develop lung cancer, so the incidence in this group will be lower. When the connection between subgroup and unusual target distribution is not immediately intuitively clear, the result of SD is a new hypothesis to be investigated by the domain experts.

2.1 Exceptional Model Mining

Exceptional Model Mining (EMM) [12, 4] can be seen as an extension of SD: instead of a single target, EMM typically selects multiple target columns. A specific kind of *interaction* between these targets is captured by the definition of a *model class*. EMM finds a subgroup to be interesting when this interaction is exceptional, as captured by the definition of a *quality measure*. For instance, when two numerical columns are selected as the targets, we can consider Pearson's correlation ρ as the model class. Quality measures for this model class could be ρ itself (to find subgroups on which the target correlation is unusually high), $-\rho$ (to find subgroups with unusually strongly negative target correlation), $|\rho|$ (to find subgroup with unusually strong positive or negative target correlation), or $-|\rho|$ (to find subgroups with unusually weak target correlation). Hence, the model class fixes the type of target interaction in which we are interested, and the quality measure fixes what, within this type of interaction, we find interesting. Several model classes have been defined and explored; for instance, Bayesian networks [5], and regression [3]. Popular quality measure for SD/EMM include WRAcc [10], z-score [13], and KL divergence [11].

2.2 Learning Behavior Modeling

Learning behavior modeling for students in MOOCs is generally aimed at predictive analytics such as dropout prediction, passing state prediction, and grades prediction. For instance, latent factors and state machines are employed to model the hidden study state of students for a predictive task [18, 16, 21]. Khajah et al. [8] integrate Latent factor and knowledge tracing with a hierarchical Bayesian model, which can consider the study skill for prediction tasks. Recurrent neural network and LSTM have been used to model study trajectories for the learning results prediction [15, 22]. Most of these existing methods focus on modeling individual behavior but do not consider the sparsity, inconsistency and heterogeneity of learning behavior data. Our methods focus on discovering exceptional learning behaviors on the subgroup level, which provide interpretable information about where the predictive model does not perform well. This allows us to establish an improved model for prediction tasks for both normal and exceptional behavioral patterns.

3. PRELIMINARIES

We assume a dataset Ω: a bag of N records $r \in \Omega$ of the form $r = (a_1, \ldots a_k, l_1, \ldots, l_m)$, where k and m are positive integers. We call $a_1, \ldots, a_k$ the *descriptive attributes* or *descriptors* of r, and $l_1, \ldots, l_m$ the *target attributes* or *targets* of r. The descriptive attributes are taken from an unrestricted domain $\mathcal{A}$. Mathematically, we define descriptions as functions $D : \mathcal{A} \to \{0, 1\}$. A description D covers a record r^i if and only if $D(a_1^i, \cdots, a_k^i) = 1$.

Definition 1. *A subgroup corresponding to a description D is the bag of records $G_D \subseteq \Omega$ that D covers, i.e.:*

$$G_D = \left\{ r^i \in \Omega \middle| D(a_1^i, \ldots, a_k^i) = 1 \right\}$$

This merely formalizes the standard LPM conditions: we seek subgroups that are defined in terms of conditions on the descriptors, hence our results are interpretable. Those conditions select a subset of the records of the dataset: those records that satisfy all conditions. These subgroups must be evaluated, which is done by the quality measure:

Definition 2. *A quality measure is a function $\varphi : \mathcal{D} \to \mathbb{R}$ that assigns a numeric value to a description D. Occasionally, we use $\varphi(G)$ to refer to the quality of the induced subgroup: $\varphi(G_D) = \varphi(D)$.*

Typically, a quality measure assesses the subgroup at hand based on some interaction on the target columns. Hence, a description and a quality measure interact through different partitions of the dataset columns; the former focuses on the descriptors, the latter focuses on the targets, and they are linked through the subgroup.

Since subgroups select subsets of the dataset at hand, and many such subsets exist, we need to employ a search strategy to ensure that we find good results in a reasonable amount of time. To do so, we employ the *beam search* algorithm as outlined in [4, Algorithm 1]. This algorithm holds the middle ground between a pure greedy search algorithm, which is likely to quickly end up in a local optimum, and an exhaustive search, which is likely to require too much time for providing the global optimum. Beam search builds up candidate subgroups in a level-wise manner, by imposing a single condition on a single attribute at each step of the search. In subsequent steps, promising candidates are *refined*, by conjoining to these candidates all possible additional single conditions on a single attribute, and evaluating the results. A purely greedy approach would, at each step, refine the single most promising candidate. By contrast, beam search refines a fixed number w (the *beam width*) of most promising candidates at each step. The larger the choice of w, the more likely we are to escape local optima, and the longer the algorithm will take. An additional parameter of beam search is the number d (the *search depth*), which sets an upper limit to the number of steps in the search process. Hence, by design, any subgroup resulting from a beam search procedure must be defined as a conjunction of at most d conditions on single attributes. The larger the choice of d, the more expressive the results are; the smaller the choice of d, the easier the results are to interpret.

4. EXCEPTIONAL LEARNING BEHAVIOR ANALYSIS

Our dataset originates from the learners involved in the EIT Digital MOOCs at Coursera. EIT Digital, as part of the

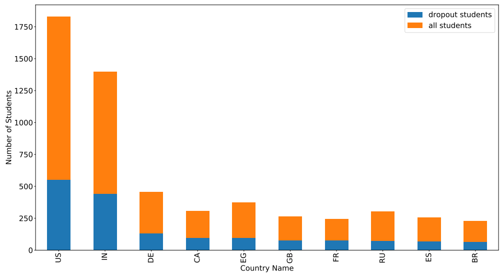


Figure 3: Dropout ratio of students by country.

Table 1: Exceptional dropout rate in subgroups. Results show subgroups with highly exceptional dropout rate. The overall dropout rate is 0.4286.

D	φ_{WRAcc}	dropout	$\lvert G_D\rvert$
Country = OM, Was Group Sponsored != True, Was Finaid Grant != True	0.0338	0.0	42
Region = MOW, Gender != male, Join Date <= 1011, Join Date > 389	0.0336	0.0	57
Country = KR, Gender != female, Profile language != ko	0.0330	0.7812	32
Country = KR, Educational status != MASTERS DEGREE, Gender != female, Was Group Sponsored != True	0.0313	0.7742	34
Country = KR, Was Group Sponsored != True	0.0304	0.7222	36

European Institute for Innovation and Technology, aims to drive Europe's digital transformation, also for education. The EIT Digital academy is focused on mobility and entrepreneurship and is at the forefront of integrating education, research, and business. The MOOCs in the online programme, have been developed by the partner universities involved in the EIT Digital Master School in Embedded Systems, in a best of breeds approach.

Together, the MOOCs form the EIT Digital online programme "Internet of Things through Embedded Systems". The online programme aims to build the reputation of EIT Digital, the partner universities, and the involved teachers. It also helps to renew pedagogy through scalable education technologies and data driven education. Learning analytics are at the core of this feedback mechanism. The online programme is comparable to an edX's micromaster and similarly offers an online equivalent of a 25 ECTS first semester; the online programme offers learners to study at their own pace, any time, any place. Moreover, they first can have a try before they commit themselves to the whole master programme. Once selected and admitted on campus, the learners can finish the double degree master programme of EIT Digital Master School in Embedded Systems.

Figure 2 displays the distributions of students across various demographic categories. In order to catch the inherent imbalance, we use demographic columns as the left hand attributes, to formulate subgroup descriptions. In the data preprocessing process, we convert the join dates, which represents how long a student has registered in Coursera, from the format of 'Datetime' to the integer days. The following three sections illustrate what kind of discoveries can be made by wielding various tools from the EMM toolbox.

Table 2: Exceptional correlation analysis between length of behavior sequence and course grades. The overall correlation coefficient ρ is 0.7406.

D	φ_{scd}	ρ	$\lvert G_D\rvert$
Country = LT, Join Date > 701, Browser language != et-EE	0.9999	0.9782	11
Region = 6	0.9994	-0.1272	10
Region = QUE	0.9992	-0.0788	11
Country = NP	0.9985	0.9630	11
Browser language = es-MX	0.9973	0.1203	7

Table 3: Exceptional classifier behavior for course passing state prediction. Results indicate that the classifier cannot work well on these exceptional subgroups.

D	φ_{f1}	$\lvert G_D\rvert$
Country = OM, Profile language = en-US, Browser language != en-US, Educational status != BACHELOR DEGREE	0.5051	32
Country = OM, Profile language != en-US	0.4058	22
Region = MA, Gender = female, Educational status=COLLEGE NO DEGREE	0.3489	24
Country = OM, Met Payment Condition != True	0.3464	31
Join Date <= 390, Region != MA	0.3193	28

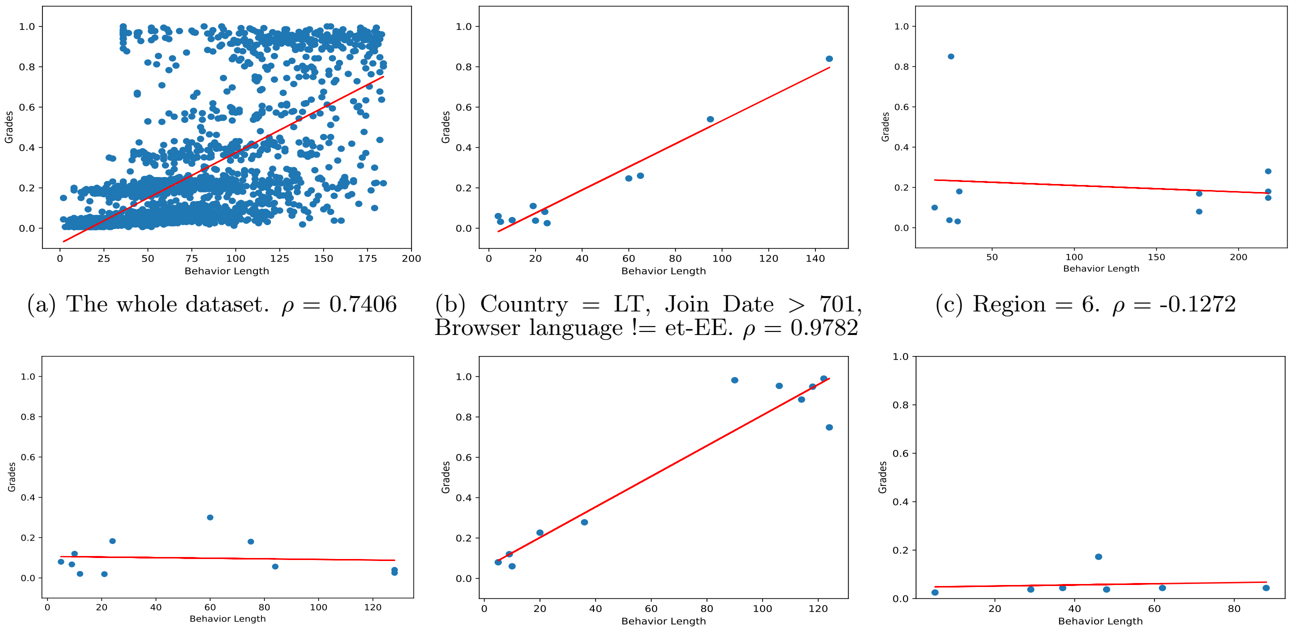
4.1 Exceptional Dropout Rate Analysis

In this section, our task is to find out the subgroups which have significantly different dropout rate compared with the whole dataset. For the purposes of this paper, we define a dropout student to be a student who has participated in at least one assessment question, but has not obtained an overall course grade. In Figure 3, we present the highest-frequency countries, and the dropout rate of students in those countries. From the figure we can see that both frequency and dropout rate vary a lot. The high dropout rate is usually seen as a defect of MOOCs. If we were to discover what kinds of students have exceptional dropout rates, then that would allow us to direct specific guidance to those students that most require it. Traditional partition and clustering methods are not qualified for this task, because they cannot provide interpretable results about the subsets of students and quantitative information about how different the subsets of students are from the whole dataset. To address this problem, we propose to engage subgroups as a partition for the whole dataset, and look for subgroups that have most exceptional dropout rate comparing with the whole dataset, employing *Weighted Relative Accuracy* (WRAcc) [20]:

$$\varphi_{\text{WRAcc}} = \frac{|G_D|}{N}\left(\frac{S_D}{|G_D|} - \frac{S_\Omega}{N}\right)$$

Here, $|G_D|$ represents the number of records covered by subgroup description D, S_D represents the number of dropout students in subgroup G_D, S_Ω represents the total number of dropout students in the whole dataset, and N represents the number of students who join this course and participated in at least one assessment question.

The beam search algorithm as described in [4, Algorithm 1] is parameterized with beam width 20 and search depth 4. The overall dropout rate is 0.4286. In Table 1, we presents the top-5 subgroups with most exceptional dropout rate. The subgroup with description "D: Region = MOW, Gender != male, Join Date between 389 and 1011" has a dropout rate



(a) The whole dataset. ρ = 0.7406

(b) Country = LT, Join Date > 701, Browser language != et-EE. ρ = 0.9782

(c) Region = 6. ρ = -0.1272

(d) Region = QUE. ρ = -0.0788

(e) Country = NP. ρ = 0.9630

(f) Browser language = es-MX. ρ = 0.1203

Figure 4: Exceptional correlations in subgroups.

of zero: all students in that subgroup complete the course. On the other hand, the subgroup with description "D: Country = KR, Gender != female and Profile language != ko", has an elevated dropout rate of 0.7812: most of these students drop out. Based on these results, we can conclude that Korean males who have set their profile language to something other than Korean, are in need of more attention. This may be a group of students who are foreigners in Korea, or Koreans who are studying in a language which is non-native to them. By identifying such at-risk groups, educators can more effectively channel their remedial activities.

4.2 Exceptional Correlation Analysis

Generally, more active students can be expected to obtain higher grades. To investigate this phenomenon, we look into the relation between the activity length (denoted by q) of students and the overall grades (denoted by g) in a course. We engage the correlation model class for EMM to realize this task. In this model class, we can estimate the correlation coefficient by calculating the sample correlation as follows:

$$\begin{aligned} \widehat{r} &= \frac{\sum(q^i - \bar{q})(g^i - \bar{g})}{\sqrt{\sum(q^i - \bar{q})^2 \sum(g^i - \bar{g})^2}} \\ z' &= \frac{1}{2}\ln\left(\frac{1+\widehat{r}}{1-\widehat{r}}\right) \\ z^* &= \frac{z' - z^C}{\sqrt{\frac{1}{|G_D|-3} + \frac{1}{|G_D^C|-3}}} \end{aligned} \tag{1}$$

Here, $\widehat{r}$ represents the sample correlation, q^i, g^i represent the activity length and course grade of each student, and $\bar{q}, \bar{g}$ represent their average values over the dataset. Equation (1) is the Fisher z transformation, z' in the lower equation represents the z' computation on the subgroup and z^C on its complement, and $|G_D|$ represents the number of records covered by subgroup with description D. Under the null hypothesis that the correlation between q and g is the same inside and outside of the subgroup, z^* follows a standard normal distribution. Hence, the value for z^* implies a p-value under this null hypothesis. Leman et al. [12] propose to use one minus this p-value as quality measure φ_{scd}: the higher this value is, the more certain we are that the null hypothesis is false and hence exceptional correlations are observed.

Using this quality measure, we conduct the experiment with beam width 20 and search depth 3. In Table 2 and Figure 4, we list the top-5 subgroups with exceptional quality score, coefficients, and coverage. We can see that some students gain extremely high grades with longer behavior sequence (cf. Figure 4b, 4e); some students have longer behavior sequence length but lower grades (cf. Figure 4c, 4d); and for some subgroups, the length of behavior sequences has no obvious correlation with the grades (cf. Figure 4f). We can deduce that the efforts that some students spend in the study are not directly correlated with their learning results.

4.3 Exceptional classifier behavior analysis

Students' behavioral records in MOOCs are sparse, inconsistent and heterogeneous. Learning behavior could be very different between different students. This imbalance increases the difficulty of training a classifier that can perform well on each part of the dataset. This makes it difficult to train a model that is qualified for tasks like dropout prediction and course passing state prediction.

In this section, we investigate whether learning behavior can predict whether or not a student can pass the course. At

Table 4: Course statistics.

course_name	course_level	complete_number	avg_grades	course_enroll_num	max_grades	min_grades	pass_number
Marketing	I	1141	0.105	4609	1	0.006	52
Design Thinking	I	369	0.167	3483	0.972	0.01	22
IoT	A	8	0.098	241	0.1	0.087	0
System Validation (2)	I	63	0.412	1010	1	0.05	12
Smart IoT	B	905	0.216	6035	1	0.004	100
Computer Architecture	I	913	0.510	7652	1	0.025	299
System Validation (4)	A	17	0.597	985	1	0.071	9
Quantitative Model (1)	I	429	0.395	1807	1	0.007	49
System Validation (3)	A	45	0.418	764	1	0.057	11
Quantitative Model (2)	A	979	0.339	4975	1	0.016	52
System Validation	I	601	0.376	2605	1	0.04	124
Technology	I	258	0.232	3930	1	0.002	34
Embedded Systems	I	549	0.291	3737	1	0.02	67
Software Architecture	A	2710	0.299	10487	1	0.012	331
Real-Time Systems	I	3615	0.203	15123	1	0.006	389
IoT Devices	I	430	0.318	6609	1	0.008	85
Embedded Hardware	I	3943	0.160	19592	1	0.02	128
Open Innovation	I	480	0.137	3150	0.969	0.008	24

the same time, we investigate in which parts of the dataset the classifier does not work well. In Section 4.1 and 4.2, we have presented that EMM can effectively discover exceptional learning behavioral patterns in MOOCs. We will continue using the EMM framework to find where our predictive model does not work well in the dataset. Considering the activities of students in assessments, forum discussions and peer assignments, we formulate the passing state prediction problem as follows:

$$f : \mathcal{X}^i \rightarrow Y^i$$

Our aim is to train a classifier f that can automatically map $\mathcal{X}^i$ to Y^i, where $\mathcal{X}^i$ is a 8-tuple $(s^i, m^i, o^i, c^i, b^i, e^i, h^i, p^i)$ feature vector representing the length of assessment and question sequence (s^i), number of assessment types (m^i), number of question types (o^i), number of correctly answered questions (c), number of asked, answered and liked questions in the forum (b^i, e^i, h^i), and peer review score (p^i), and where Y is the label of passing state: $\{0, 1\}$. We normalize the features into 0 to 1 as the input values.

At first, the classifier is trained on the whole dataset. This model will classify some students correctly and some students wrongly; in any case we find a value of predicted labels $\hat{Y}$. These two binary values Y and $\hat{Y}$ will agree and disagree on some students, and that interaction can be used to capture the quality of the classifier predictions in a single number. We use the f1 score to capture this:

$$\varphi_{\text{f1}} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2}$$

However, we can perform the exact same computation for a subset of the vectors Y and $\hat{Y}$, for instance the subset induced by a subgroup. Thus, we employ φ_{f1} as a quality measure for EMM.

We conduct the experiment by setting the search depth to 4 and beam width to 10. We engage an SVM classifier as the predictive model[1], which has 0.85 as f1 score on the whole dataset. In Table 3 we list the top-5 subgroups with exceptional behavior. We can see that even though the classifier performs well on the whole dataset, in some subgroups it does not. Particularly for the students described by descriptions like "D: Region = MA, Gender = female, Educational status=COLLEGE NO DEGREE", the classifier performs poorly on the prediction task at hand: the support vector machine has trouble predicting the study success of Massachusets women without a college degree. Hence, this group requires a more sophisticated classifier.

5. CONCLUSIONS

In this paper, we employ Exceptional Model Mining (EMM) for exceptional learning behavior analysis in MOOCs. Rather than predicting the success of individual students, which is difficult due to the inherent sparsity, inconsistency, and heterogeneity of the data, EMM specializes in identifying coherent groups that behave differently from the norm. Since the subgroups resulting from EMM come with an easily interpretable definition, Exceptional Model Mining allows educators to more effectively channel their remedial activities.

We employ three EMM model classes for different tasks of learning behavior analysis. Experimental results on a real Coursera dataset show that for some students, the dropout rate is very different from the whole dataset, the learning efforts are not always correlated with course grades, and a classifier that performs very well on the whole dataset has trouble on some subpopulations of the data. In future work, we will make use of these discovered exceptional behavioral patterns to establish an improved model, which can model both normal and exceptional learning behaviors for the students in MOOCs. We plan to develop a modeling method that can perform well on each part of the dataset, including the exceptional ones.

6. ACKNOWLEDGMENTS

This research was funded by EIT 18008-A1803 project. In addition, Xin Du would like to thank China Scholarship Council (CSC) for the financial support.

[1] one may plug in one's preferred classifier; SVM selection is merely meant as an illustration, not an endorsement.

References

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A. I. Verkamo. Fast Discovery of Association Rules. *Advances in Knowledge Discovery and Data Mining*, pp. 307–328, 1996.

[2] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[3] W. Duivesteijn, A. Feelders, and A. Knobbe. Different slopes for different folks: mining for exceptional regression models with cook's distance. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 868–876, 2012.

[4] W. Duivesteijn, A. J. Feelders, and A. Knobbe. Exceptional model mining. *Data Mining and Knowledge Discovery*, 30(1):47–98, 2016.

[5] W. Duivesteijn, A. Knobbe, A. Feelders, and M. van Leeuwen. Subgroup discovery meets Bayesian networks — an exceptional model mining approach. In *10th International Conference on Data Mining (ICDM)*, pp. 158–167, 2010.

[6] D. Hand, N. Adams, R. Bolton (eds). *Pattern Detection and Discovery.* Springer, New York, 2002.

[7] F. Herrera, C. J. Carmona, P. González, and M. J. Del Jesus. An overview on subgroup discovery: foundations and applications. *Knowledge and Information Systems*, 29(3):495–525, 2011.

[8] M. Khajah, R. Wing, R. Lindsey, and M. Mozer. Integrating latent-factor and knowledge-tracing models to predict individual differences in learning. In *Educational Data Mining*, 2014.

[9] W. Klösgen. Explora: A Multipattern and Multistrategy Discovery Assistant. *Advances in Knowledge Discovery and Data Mining*, pp. 249–271, 1996.

[10] M. van Leeuwen and A. J. Knobbe. Non-redundant subgroup discovery in large and complex data. In *Proceedings of the European Conference on Machine Learning & Principles and Practice of Knowledge Discovery in Databases*, pp. 459–474, 2011.

[11] M. van Leeuwen and A. J. Knobbe. Diverse subgroup set discovery. *Data Mining and Knowledge Discovery*, 25(2):208–242, 2012.

[12] D. Leman, A. Feelders, and A. Knobbe. Exceptional model mining. In *Proceedings of the European Conference on Machine Learning & Principles and Practice of Knowledge Discovery in Databases*, pages 1–16. Springer, 2008.

[13] M. Mampaey, S. Nijssen, A. Feelders, R. Konijn, and A. Knobbe. Efficient algorithms for finding optimal binary features in numeric and nominal labeled data. *Knowledge and Information Systems*, 42(2):465–492, 2015.

[14] K. Morik, J. F. Boulicaut, A. Siebes (eds). *Local Pattern Detection.* Springer, New York, 2005.

[15] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pp. 505–513, 2015.

[16] J. Qiu, J. Tang, T. X. Liu, J. Gong, C. Zhang, Q. Zhang, and Y. Xue. Modeling and predicting learning behavior in moocs. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pp. 93–102, 2016.

[17] M. Qiu, F. Zhu, and J. Jiang. It is not just what we say, but how we say them: Lda-based behavior-topic model. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pp. 794–802, 2013.

[18] A. Ramesh, D. Goldwasser, B. Huang, H. Daume III, and L. Getoor. Learning latent engagement patterns of students in online courses. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[19] 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.

[20] L. Todorovski, P. Flach, and N. Lavrač. Predictive performance of weighted relative accuracy. In *Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery*, pages 255–264, 2000.

[21] F. Wang and L. Chen. A nonlinear state space model for identifying at-risk students in open online courses. *Proceedings of the 9th International Conference on Educational Data Mining*, pp. 527–532, 2016.

[22] L. Wang, A. Sy, L. Liu, and C. Piech. Learning to represent student knowledge on programming exercises using deep learning. In *Proceedings of the 10th International Conference on Educational Data Mining*, pp. 324–329, 2017.

[23] S. Wrobel. An Algorithm for Multi-relational Discovery of Subgroups. In *Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 78–87, 1997.

[24] Z. Zhao, Z. Cheng, L. Hong, and E. H. Chi. Improving user topic interest profiles by behavior factorization. In *Proceedings of the 24th International Conference on World Wide Web*, pp. 1406–1416, 2015.

Towards a Model-Free Estimate of the Limits to Student Modeling Accuracy

Binglin Chen
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
chen386@illinois.edu

Matthew West
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
mwest@illinois.edu

Craig Zilles
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
zilles@illinois.edu

ABSTRACT

This paper attempts to quantify the accuracy limit of "next-item-correct" prediction by using numerical optimization to estimate the student's probability of getting each question correct given a complete sequence of item responses. This optimization is performed without an explicit parameterized model of student behavior, but with the constraint that a student's likelihood of getting a problem correct only increases or remains unchanged with additional practice (i.e., no forgetting). We present results for this method for the Assistments 2009–2010 data where it suggests that there is only modest opportunity for improvement beyond the state of the art predictors. Furthermore, we describe a framework for applying this method to datasets where problems can be tagged with multiple skills and problem difficulties. Lastly, we discuss the limitations of this method, specifically its inability to give tight bounds on short sequences.

1. INTRODUCTION

Student modeling is a fundamental building block of educational systems that are intelligent or adaptive. With a model of a student, such a system can consider all of the actions it has available and make a prediction about which ones are likely to be the most profitable for a particular student at the current time.

One class of student models tries to predict *next-item-correct*, i.e., what is the probability that a student's attempt on the next item presented will be correct given the student's results on all previous items. For a number of years, this topic saw vigorous research with non-trivial improvements using improved model parameterizations [1, 6, 7, 11] and recurrent neural networks [10]. Yet, performance of next-item-correct predictors has seemed to reach an asymptote that is far below perfect prediction.

This gap between the current state of the art and perfect prediction raises the question of how much headroom remains for further improvements to next-item-correct prediction. Previous work by Beck and Xiong [2] has attempted to characterize the accuracy limit by analyzing the performance of a collection of "cheating" prediction algorithms that employ a partial knowledge of future results. They conclude that further large improvements in prediction accuracy are unlikely.

Estimating a tight bound to prediction accuracy is challenging, because one needs to utilize some information about future correctness without merely regurgitating the stream of actual outcomes as one's predictions, which would yield the tautological bound of 100% accuracy. Beck and Xiong navigate this conundrum by allowing their cheating model to correctly predict the transitions from giving an incorrect response to giving a correct response (e.g., learning), but not those from giving an correct response to giving a incorrect response (in their words, "forgetting"). We found this approach to be unsatisfying in two respects. First, the time period in which the data is collected is too short for true forgetting to take place, it is rather more likely to be slipping, so we feel that the model is a mismatch for the phenomena at hand. Second, we feel that perfectly predicting incorrect-to-correct transitions but not correct-to-incorrect transitions seems arbitrary.

Instead, we posit that the limits of accuracy for next-item-correct prediction derive from the fact that learning is not a binary transition from a state of not knowing to a state of knowing, but rather that there is a continuum of knowledge levels that a student could be at. For example, there is a point on this continuum where a student will get 50% of the problems attempted correct and the other 50% incorrect. The challenge for next-item-correct prediction for such a student is precisely determining whether the next attempt will be correct or incorrect, much like the hopeless task of trying to consistently predict the outcome of flipping a fair coin. More precisely, it is the student responses as they transition from not knowing to knowing that are hard to predict, as the behavior of perfectly knowledgeable and perfectly unknowledgeable students is trivial to predict.

Thus, the limit for prediction should primarily derive from the fraction of a data stream during which students are in this transitional phase where they are intermingling correct and incorrect responses. This can be viewed as the amount of entropy in the data, and this entropy can and does vary from dataset to dataset. As such, we believe that a method

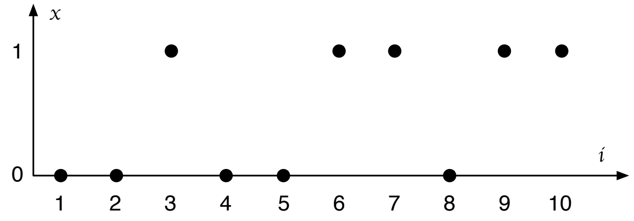


Figure 1: Illustrative example of the input to the next-item-correct prediction problem. For this example, $n = 10$ and $x_1, \ldots, x_n = 0, 0, 1, 0, 0, 1, 1, 0, 1, 1$.

that can estimate the limits of predictability as a function of this entropy can serve as a less arbitrary estimate of the accuracy limit for next-item-correct prediction and serve as a useful means for characterizing and comparing datasets.

This paper is organized as follows. We first formalize the next-item-correct prediction problem in Section 2. We then describe our model-free bounding method in Section 3. We show experimental results of our method in Section 4. Finally, we discuss the limitations of our method in Section 5 and future directions in Section 6.

2. NEXT-ITEM-CORRECT PREDICTION

We formalize the next-item-correct prediction problem as follows. We are given a length-n sequence $x_1, \ldots, x_n$, where $x_i = 1$ if the student answered the ith attempted item correctly and $x_i = 0$ otherwise, as shown in Figure 1. Given this information, we want to produce n reals $p_1, \ldots, p_n$ where p_i is the probability of the student answering the ith attempted item correctly. Typically models are required to produce $p_1, \ldots, p_n$ in order and they are only allowed to look at $x_1, \ldots, x_{t-1}$ when producing p_t, as future observations should not be available during prediction. Some of the notable models for this task are Bayesian Knowledge Tracing (BKT) [3], Performance Factor Analysis (PFA) [8], and Deep Knowledge Tracing (DKT) [10].

In efforts to improve their performance, many models use the *knowledge components* required by each item, denoted as $\vec{s}_1, \ldots, \vec{s}_n$. Each $\vec{s}_i$ is a d dimensional vector where d is the number of knowledge components in the corresponding dataset. Each entry of $\vec{s}_i$ is typically boolean, indicating whether the item requires the corresponding knowledge component. The entries of $\vec{s}_i$ can be real valued as well, indicating the degree of mastery required on each component in order to answer the item correctly.

With the ground truth $x_1, \ldots, x_n$ and predictions of a model $p_1, \ldots, p_n$, a performance metric $\mathcal{L}$ is typically used to measure how good the predictions are. The most widely used metrics for this task are root mean squared error (RMSE) and area under the curve (AUC) [9]. Log likelihood (LL) has also been proposed [9] though it has not been widely used on this task. This paper will use average LL instead of LL since the former does not depend on the size of the data. Models with better $\mathcal{L}(p_1, \ldots, p_n; x_1, \ldots, x_n)$ are to be preferred. The meaning of "better" depends on the metric; larger values are better for average LL and AUC while smaller values are better for RMSE.

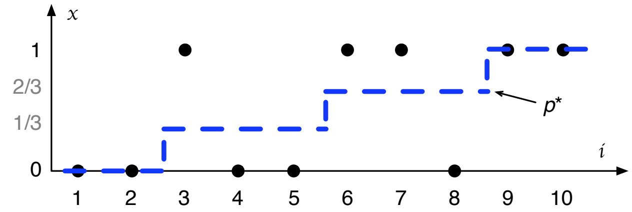


Figure 2: Results of the model-free bounding method when all items require the same knowledge component.

3. MODEL-FREE ACCURACY BOUNDS

The core idea of our method is that the probability of a student correctly answering items that require the same knowledge components should be non-decreasing over the short term. More precisely, if the current item is no more difficult than a previous item that requires the same knowledge and there hasn't been sufficient time or interference for forgetting to occur, the student's probability of getting the current item correct should be at least as high as the previous item.

This idea is illustrated in Figure 2, where the dashed line segments correspond to the probability of the student correctly answering each item. One could interpret this sequence as having three phases: (1) items 1 and 2 as a region of unknowing where the student gets every item incorrect, (2) items 3 through 8 as a region of learning where correct and incorrect responses are interleaved, and (3) items 9 and 10 as a region of mastery where the student gets every item correct. Even though the second region includes both correct and incorrect responses, we are interpreting those merely as events from an underlying probability distribution and that probability of correct responses is non-decreasing throughout the sequence.

Based on this idea, our proposed bounding method finds correctness probabilities for each item $p^*_1, \ldots, p^*_n$ that optimize $\mathcal{L}(p^*_1, \ldots, p^*_n; x_1, \ldots, x_n)$ subject to the constraint that the p^*_i sequence is non-decreasing on appropriate item sequences. These p^*_i provide the best local estimate of the likelihood that a student will get an item correct given an assumption that only learning is occuring. To do better, one would have to predict the precise sequence of correct and incorrect responses and we believe that this problem is akin to predicting the precise sequence of heads and tails from repeated flips of a coin. As such, we expect this to be a practical bound to next-item-correct prediction.

We refer to this method as being "model free", because it does not rely on any parameterized model of student behaviors and does not require training. Instead, the p^*_i values are derived directly from the sequence $x_1, \ldots, x_n$ and, therefore, can be potentially applied on any dataset.

3.1 Single knowledge component case

Before diving into the case where multiple knowledge components are involved, we first explain our method in the simplest case where the sequence of items require the same knowledge component. In this case, since all of the items are equivalent in terms of knowledge components, the aforementioned constraint is equivalent to constraining $p_1, \dots, p_n$ to be non-decreasing. Thus our method reduces to solving the following numerical optimization problem to obtain $p_1^*, \dots, p_n^*$:

$$\begin{aligned} \text{optimize: } & \mathcal{L}(p_1, \dots, p_n; x_1, \dots, x_n) \\ \text{subject to: } & 0 \le p_i \le 1 \text{ for all } i \\ & p_i \le p_j \text{ for all } i < j. \end{aligned} \tag{1}$$

This numerical optimization problem can be solved efficiently by an interior point method if (1) $\mathcal{L}$ is convex and smaller $\mathcal{L}$ is better, or (2) $\mathcal{L}$ is concave and larger $\mathcal{L}$ is better. Out of the three metrics mentioned previously, average LL and RMSE satisfy this criterion while AUC is not even continuous (and hence not convex or concave). Thus this formulation as a numerical optimization problem is only applicable when $\mathcal{L}$ is average LL or RMSE. There are various tools that can solve this sort of numerical optimization problem. In our implementation we used used Matlab's *fmincon* with L-BFGS as the Hessian method.

To give a sense of what this method produces, Figure 2 shows as the dashed line the values $p_1^*, \dots, p_n^*$ that minimize RMSE for the given observed item responses $x_1, \dots, x_n$ (solid black dots).

3.2 Partial order of items

In order to handle sequences of items with different combinations of multiple knowledge components, we need to be able to compare the items and decide which previously attempted items provide information useful for predicting the outcome of the current item. The intuition is that if item a is the same difficulty or easier with respect to the required knowledge components than item b, then a student should do item a at least as well as item b. We compare items by defining a partial order $\preceq$ over the knowledge component vectors as follows:

$$\vec{s}_a \preceq \vec{s}_b \iff \vec{s}_{a,k} \le \vec{s}_{b,k} \text{ for all } k, \tag{2}$$

where $\vec{s}_{a,k}$ is the kth coordinate of $\vec{s}_a$. This partial order essentially states that item a should be considered easier than or equal to item b if the required mastery level of each knowledge component of item a is less than or equal to that of item b. Intuitively, given $\vec{s}_a \preceq \vec{s}_b$, then a student should be able to answer item a correctly if the student can answer item b correctly.

Given this definition of partial order, we can induce a directed acyclic graph (DAG) on the set of items, where there is an edge from the jth item to the ith if and only if $i < j$ and $\vec{s}_j \preceq \vec{s}_i$. The intuition of the requirement $i < j$ is that being able to solve a "harder" item in the past implies being able to solve an "easier" item in the future. However, being able to solve a "harder" item in the future does not imply being able to solve an "easier" item in the past since the student might have learned a lot in between. To illustrate this, we show the DAG induced by a sequence of 6 items with 3 knowledge components in Figure 3. In such a DAG, an edge from the jth item to the ith means that the student should be able to do the jth item at least as well as the ith item.

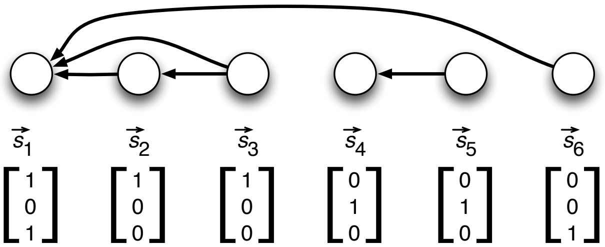

$\vec{s}_1$
$\vec{s}_2$
$\vec{s}_3$
$\vec{s}_4$
$\vec{s}_5$
$\vec{s}_6$
$\begin{bmatrix}1\\0\\1\end{bmatrix}$
$\begin{bmatrix}1\\0\\0\end{bmatrix}$
$\begin{bmatrix}1\\0\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
$\begin{bmatrix}0\\0\\1\end{bmatrix}$

Figure 3: A directed acyclic graph induced by the partial order. An arrow from the jth item to the ith item means that the student should do the jth item at least as well as the ith item. There are two connected components in this induced graph, which are $\{x_1, x_2, x_3, x_6\}$ and $\{x_4, x_5\}$.

3.3 Multiple knowledge components case

Given the partial order on items as described above, we can generalize the non-decreasing constraints for a single knowledge component to handle any combination of knowledge components. Specifically, given $i < j$ and $\vec{s}_j \preceq \vec{s}_i$, the probability p_j of the student answering the jth item correctly should not be lower than the probability p_i of the ith item since the jth item is no harder than the ith item. That is, $p_i \le p_j$ when there is an edge from the jth item to the ith item in the induced DAG on the sequence. Thus the optimization problem can be reformulated as

$$\begin{aligned} \text{optimize: } & \mathcal{L}(p_1, \dots, p_n; x_1, \dots, x_n) \\ \text{subject to: } & 0 \le p_i \le 1 \text{ for all } i \\ & p_i \le p_j \text{ for all } i < j \text{ that satisfy } \vec{s}_j \preceq \vec{s}_i. \end{aligned} \tag{3}$$

This complicated optimization problem can usually be broken down into smaller ones by dividing the sequence $x_1, \dots, x_n$ into shorter subsequences based on the connected components they belong to in the induced DAG. In the example depicted by Figure 3, there are two connected components which correspond to $\{x_1, x_2, x_3, x_6\}$ and $\{x_4, x_5\}$. We can then optimize on each subsequence separately.

Another trick to accelerate the optimization is removing redundant constraints since the partial order is transitive. For example, the constraint corresponding to the edge from $\vec{s}_3$ to $\vec{s}_1$ in Figure 3 can be safely removed since it is implied by constraints corresponding to $\vec{s}_3 \preceq \vec{s}_2$ and $\vec{s}_2 \preceq \vec{s}_1$.

3.4 Metrics that cannot be directly optimized

As mentioned before, our method is not applicable to AUC since it is not continuous. To compute a bound for AUC, we first solve the optimization problem by either maximizing average LL or minimizing RMSE. Once we obtained p_i^* for the entire dataset, we can calculate AUC using these p_i^*.

In general, we can always optimize on one metric $\mathcal{L}$ for p_i^* and evaluate the p_i^* with any metric $\mathcal{L}'$ even though the optimization is done with respect to $\mathcal{L}$. We refer to this as the bound obtained by optimizing $\mathcal{L}$.

4. EXPERIMENTAL RESULTS

We applied BKT, DKT, and our method to the Assistments 2009–2010 dataset. We chose this dataset because it has

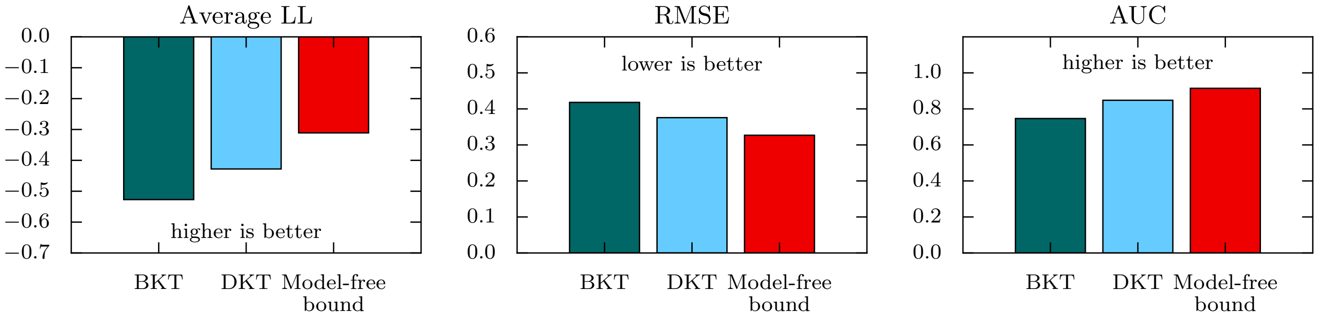


Figure 4: Results of applying BKT, DKT, and our method to Assistments 2009–2010 dataset.

relatively long sequences of attempts. We used the same train/test split for this dataset as in Khajah et al. [5]. We used the BKT implementation by Yudelson[1] [11] with the default parameters and Baum-Welch as the training method. We used Khajah et al.'s [5] implementation of DKT[2] with default parameters. We only applied our method to the test set for meaningful comparisons.

For the rest of this paper, we only report bounds obtained by maximizing average LL. Throughout our experiments, we found that the bounds for all of average LL, RMSE, and AUC obtained by minimizing RMSE differed by less than 0.5% from those obtained by maximizing average LL. In fact, it can be proved that minimizing RMSE and maximizing average LL will yield the same p_i^* in the single knowledge component case (Equation 1). See the Appendix for the proof.

We show our results on Assistments 2009–2010 for average LL, RMSE, and AUC in Figure 4. The performance of DKT is roughly half way between BKT and the bound produced by our method for all of the metrics. This suggests that the room for further improvements on Assistments 2009–2010 is limited.

5. LIMITATIONS

The major limitation of our method is its optimistic nature, meaning that it can produce a bound that is too loose. This optimism manifests in two ways: first, our method can predict the precise location of learning transitions, which will be difficult for any realistic model, and, second, more generally when the sequence of predictions to be made is short the model isn't significantly constrained.

5.1 Predicting Particular Events

The proposed technique appears to provide a reasonable bound of prediction performance when student behavior follows a non-instantaneous learning of a topic involving an interleaving of correct and incorrect responses as shown in Figure 1. However, when students transition instantly from consistently answering incorrectly to consistently answering correctly, the model will likely produce a bound that is too loose. Consider the item response sequences of two students shown in Figure 5. Both of these students only transition from incorrect responses to correct responses, meaning that the optimization is free to generate predictions that precisely match the data, resulting in 100% accuracy. A real model, however, must predict the point of the transition, knowing that after observing the first three incorrect responses it should predict correct for the first student's fourth attempt and incorrect for the second student's fourth attempt. While it isn't impossible to imagine that there are features to guide such a prediction, it is difficult to believe that it could be done consistently with 100% accuracy.

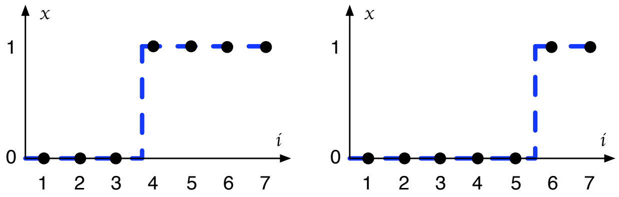


Figure 5: Two sequences that our method predicts perfectly. A real predictor, however, might have trouble predicting the precise location of the upward transition.

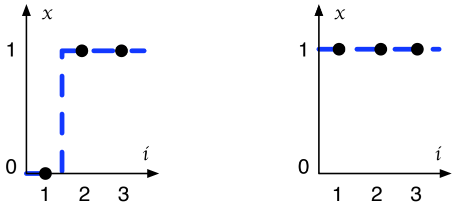


Figure 6: Our method can predict initial behavior perfectly in some circumstances. A real predictor, however, might have trouble predicting precisely which students would get a problem correct on their first attempt.

A special case of predicting such a transition is predicting whether or not the very first attempt is going to be correct. As shown in Figure 6, our method can perfectly predict whether or not a student gets their first attempt correct, provided the student gets all other attempts correct. A real system might be challenged to predict precisely which students would perform in this manner, although some knowl-

[1] https://github.com/IEDMS/standard-bkt
[2] https://github.com/mmkhajah/dkt

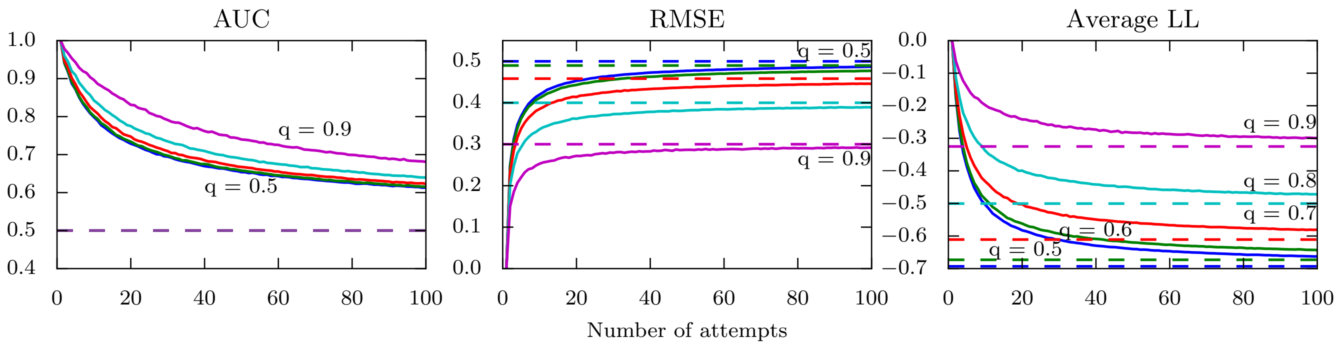


Figure 7: Upper bounds produced by our method versus theoretical bounds for attempt results that are i.i.d. with fixed q for various sequence lengths. The solid curves correspond to the results of our method and the dashed lines correspond to the theoretical bounds.

edge about the students will certainly enable such predictions to be performed at a rate better than just the average frequency that students get a given question correct on their first attempt. Nevertheless, these features of the data lead our system to be optimistic, and these features occur more frequently and have larger impact on short sequences.

5.2 Short Sequences

In general, our method struggles with short sequences, because the optimization is largely unconstrained. For example, consider the case where every student has made exactly one attempt. In such a case our method will always produce p_1^* that is exactly the same as x_1, which results in a trivial bound of 100% accuracy. However, as the sequence length increases, the constraints will generally prevent our method from being perfectly accurate, and thus it will provide a more useful bound.

To understand how the amount of optimism in our method depends on the sequence length, we used independent and identically distributed (i.i.d.) coin tosses to study this. Such sequences allow us to compute a theoretical bound that we can compare to the one produced by our method. When attempt results $x_1, \ldots, x_n$ are i.i.d. with probability q of being correct, the theoretical bound is $q \log q + (1-q) \log(1-q)$ for average LL, $\sqrt{q(1-q)^2 + (1-q)q^2}$ for RMSE, and 0.5 for AUC.

Specifically, we generated i.i.d. results with sequence lengths ranging from 1 to 100 and with q ranging from 0.1 to 0.9 and same $\vec{s}$ for every attempt. For each length, we generated 10,000 sequences and computed the bound for average LL, RMSE, and AUC using our method.

We plotted the bounds computed by our method and the theoretical bound in Figure 7. We chose to not plot the results for q from 0.1 to 0.4 in the figure since we found that q and $1-q$ yield the same results. The solid curves in the figure correspond to the results of our method for each q while the dashed lines correspond to the theoretical bound for each q. As the figure shows, our method starts off wildly optimistic when the sequence length is 1 and gradually converges to the theoretical bounds as the sequence length increases. At a sequence length of 100, the bounds by our method are close to the theoretical bound for average LL and RMSE but not AUC. These trends suggest that our method works reasonably well for average LL and RMSE when the sequence length is large enough, however it is too optimistic on AUC even with long sequences.

6. DISCUSSION AND CONCLUSION

In this paper, we presented a model-free bounding method to find the limit of the next-item-correct prediction task. The method assumes that forgetting is absent and uses the constraint that the probability of students correctly answering a set of similar items should not decrease as they practice more. We applied our method to the Assistments 2009–2010 dataset and found that DKT's performance on this dataset is fairly close to the bound produced by our method. This suggests that the room for improvement on this dataset is small.

The main shortcoming of our method is its optimistic nature. In other words, our method will produce a bound that is too loose, especially for short sequences. While we can conceive of many ways to potentially compensate for this optimism (motivated by the scenarios discussed in Section 5), we fear that any attempts we make to estimate compensation factors has the potential to yield a result that no longer serves as a bound (i.e., that a real implementation could potentially achieve a performance exceeding our "bound"). Furthermore, we view the parameter-free simplicity of our method to be one of its virtues, and it is not clear how to preserve that while introducing such compensation. The other shortcoming is that our method does not incorporate forgetting by default. However, this could potentially be incorporated by relaxing constraints when forgetting is suspected to have occurred.

The intuition behind our method is based on the reason why next-item-correct prediction is feasible. Since independent identically distributed (i.i.d.) coin tosses are inherently unpredictable, next-item-correct prediction is feasible only if there are regularities in the data. Learning is undoubtedly the most important regularity that we would like to observe in any educational system. Thus the difficulty of the next-item-correct prediction task depends on how much students' performance deviates from i.i.d. and shows non-decreasing behavior. Our method tries to capture such regularities due to learning.

7. ACKNOWLEDGEMENTS

This work was partially supported by NSF DUE-1347722, NSF CMMI-1150490, and the College of Engineering at the University of Illinois at Urbana-Champaign under the Strategic Instructional Initiatives Program (SIIP). The authors would like to thank Luc Paquette for useful discussions.

APPENDIX

To prove that minimizing RMSE is equivalent to maximizing average LL in the case of Equation 1, we first recall the concept of a *scoring rule* [4], which is a function that scores a predictive probability distribution P against an observation x_i drawn from a target probability distribution Q that we are trying to recover. In this context a larger score indicates a better P. In the case of binary variables with range $\{0,1\}$, both P and Q are Bernoulli distributions and a scoring rule can be simply denoted as $S(p,x)$, where p is the probability of observing 1 in P and x is an observation drawn from Q.

A *strictly proper scoring rule* is a scoring rule such that the expected score over a set of observations drawn from Q is uniquely maximized when $P = Q$ [4]. The *quadratic score* and the *logarithmic score* are two commonly used strictly proper scoring rules. In the case of Equation 1, maximizing the quadratic score is equivalent to minimizing RMSE and maximizing the logarithmic score is equivalent to maximizing average LL.

In the binary case, a strictly proper scoring rule $S(p,x)$ has the Savage representation $S(p,x) = G(p) + G^*(p)(x-p)$ where G is strictly convex and G^* is a subdifferential of G [4]. Define the cost function $F(p; x_1, \ldots, x_n)$ by $F(p) = \frac{1}{n}\sum_{i=1}^n S(p, x_i) = G(p) + G^*(p)(\bar{x}-p)$ where $\bar{x} = \frac{1}{n}\sum_{i=1}^n x_i$.

Lemma 1. *$F(p)$ has a unique maximum at $p = \bar{x}$ and is strictly quasiconcave, thus unimodal.*

Proof. First observe that $F(p) = G(p) + G^*(p)(\bar{x}-p) \leq G(\bar{x}) = F(\bar{x})$ by the definition of the subdifferential, with equality if and only if $p = \bar{x}$. Thus $p = \bar{x}$ is the unique maximum.

To establish quasiconcavity, we will show that for any $\alpha \in (0,1)$, $F(\alpha p + (1-\alpha)q) > \min\{F(p), F(q)\}$. Let $r = \alpha p + (1-\alpha)q$ and, without loss of generality, assume $p < q$, so either $p < r \leq \bar{x}$ or $\bar{x} \leq r < q$. In the first case:

$$\begin{aligned}
F(r) - F(p) &= G(r) - G(p) + G^*(r)(\bar{x}-r) - G^*(p)(\bar{x}-p) \\
&> G^*(p)(r-p) + G^*(r)(\bar{x}-r) - G^*(p)(\bar{x}-p) \\
&= (G^*(r) - G^*(p))(\bar{x}-r) \\
&\geq 0.
\end{aligned}$$

The last step is due to monotonicity of G^*, which states that $(G^*(r)-G^*(p))(r-p) \geq 0$, and because $(\bar{x}-r)$ has the same sign as $(r-p)$ we have $(G^*(r)-G^*(p))(\bar{x}-r) \geq 0$. This establishes that $F(r) > F(p)$ in the first case. Similarly, $F(r) > F(q)$ in the second case, thus $F(r) > \min\{F(p), F(q)\}$. □

For any solution to Equation 1, we can partition $p_1, \ldots, p_n$ into blocks (subsets) where each member of a block has equal value and no two blocks share a value. Because Equation 1 requires monotonicity, each block must have consecutive indices.

Lemma 2. *If $\mathcal{L}$ is a strictly proper scoring rule, then every solution to Equation 1 consists of blocks of the form $p_i = \ldots = p_j = \overline{\{x_i, \ldots, x_j\}} = \sum_{k=i}^{j} x_k/(j-i+1)$.*

Proof. Consider any block $p = p_i = \ldots = p_j$ in a solution to the optimization problem described by Equation 1 when $\mathcal{L}$ is a strictly proper scoring rule. Because blocks have distinct values, p is locally unconstrained and so Lemma 1 implies $p = \overline{\{x_i, \ldots, x_j\}}$. □

Algorithm 1

```
1: $i \leftarrow 1$
2: while $i \leq n$ do
3:     find the largest $j$ with $i \leq j \leq n$ that minimizes
       $\overline{\{x_i, \ldots, x_j\}}$
4:     $p_i, \ldots, p_j \leftarrow \overline{\{x_i, \ldots, x_j\}}$
5:     $i \leftarrow j + 1$
6: end while
```

Theorem 1. *If $\mathcal{L}$ is a strictly proper scoring rule, then Algorithm 1 gives the unique solution to Equation 1.*

Proof. Let $p_1^*, \ldots, p_n^*$ be the output of Algorithm 1. Assume that $p_1, \ldots, p_n$ is a distinct solution to Equation 1. Let k be the first index for which $p_k^* \neq p_k$ and let $p_i^*, \ldots, p_j^*$ be the block with $i \leq k \leq j$.

If $p_k < p_k^*$, then monotonicity implies $k = i$. Let $\{p_k, \ldots, p_\ell\}$ be the following block, so $p_k^* > p_k = \overline{\{x_k, \ldots, x_\ell\}}$, which contradicts Line 3 in Algorithm 1.

If $p_k > p_k^*$, then $p_k^* < p_k \leq \overline{\{x_k, \ldots, x_j\}}$ because the optimization subproblems for blocks in $\{p_k, \ldots, p_j\}$ are locally unconstrained below. But by Lemma 2 we have:

$$\begin{aligned}
p_k^* &= \overline{\{x_i, \ldots, x_j\}} \\
&= \frac{k-i}{j-i+1}\overline{\{x_i, \ldots, x_{k-1}\}} + \frac{j-k+1}{j-i+1}\overline{\{x_k, \ldots, x_j\}} \\
&> \frac{k-i}{j-i+1}p_k^* + \frac{j-k+1}{j-i+1}p_k^* \\
&= p_k^*,
\end{aligned}$$

which is again a contradiction.

Note that Algorithm 1 does not depend on $\mathcal{L}$, so all strictly proper scoring rules give the same solution to Equation 1. □

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 *International Conference on Intelligent Tutoring Systems*, pages 406–415. Springer, 2008.

[2] J. Beck and X. Xiong. Limits to accuracy: how well can we do at student modeling? In *Educational Data Mining*, 2013.

[3] 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.

[4] T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

[5] M. Khajah, R. V. Lindsey, and M. C. Mozer. How deep is knowledge tracing? In *Educational Data Mining*, 2016.

[6] M. Khajah, R. Wing, R. Lindsey, and M. Mozer. Integrating latent-factor and knowledge-tracing models to predict individual differences in learning. In *Educational Data Mining 2014*. Citeseer, 2014.

[7] Z. A. Pardos and N. T. Heffernan. Kt-idem: introducing item difficulty to the knowledge tracing model. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 243–254. Springer, 2011.

[8] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis–a new alternative to knowledge tracing. In *The 14th International Conference on Artificial Intelligence in Education*, 2009.

[9] R. Pelánek. Metrics for evaluation of student models. *Journal of Educational Data Mining*, 7(2):1–19, 2015.

[10] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, pages 505–513, 2015.

[11] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In *International Conference on Artificial Intelligence in Education*, pages 171–180. Springer, 2013.

Standard Error Considerations on AFM Parameters

Guillaume Durand
National Research Council
Canada
100 rue des Aboiteaux
Moncton, NB, Canada
Guillaume.Durand@nrc.ca

Cyril Goutte
National Research Council
Canada
1200 Montreal Rd
Ottawa, ON, Canada
Cyril.Goutte@nrc.ca

Serge Léger
National Research Council
Canada
100 rue des Aboiteaux
Moncton, NB, Canada
Serge.Leger@nrc.ca

ABSTRACT

Knowledge tracing is a fundamental area of educational data modeling that aims at gaining a better understanding of the learning occurring in tutoring systems. Knowledge tracing models fit various parameters on observed student performance and are evaluated through several goodness of fit metrics. Fitted parameter values are of crucial interest in order to diagnose learning mastery as well as knowledge models and qualitative aspects of the learning environment. Unfortunately, parameter values are rarely associated with standard errors or confidence intervals, both of which are critical information to validate the inferences that can be made from the model. Taking the example of the Additive Factor Model, we describe how to obtain standard errors on the model parameters. We propose two methods to compute those and discuss results obtained on a public dataset.

Keywords

Parameters standard error, Additive Factor Model

1. INTRODUCTION

Educational Data Mining (EDM) has already produced numerous predictive models to accurately detect, anticipate and measure meaningful outcomes of learning activities. Predicting student performance has been available for years. For instance, it was the goal of the Knowledge Discovery and Data mining (KDD) Cup 2010 [1], where teams around the world competed to get the most accurate predictions on student test item successes. While predictive accuracy and overall model goodness of fit remain central concerns, others considerations have since emerged in the EDM scientific community. Model usefulness is one of them. A model can be accurate in its predictions but useless to provide additional educational values in a learning environment [10]. Another concern, of even greater interest for the work presented in this paper, is the identifiability of the models produced and used by the EDM community. The cognitive models we use for knowledge tracing are validated towards their predictive quality but their prediction performance is not necessarily where they are most useful. This is the case, for instance, for the Additive Factor Model (AFM) [3] or the Bayesian Knowledge Tracing model (BKT) [5]. Both are widely used in intelligent tutoring systems to detect when a student has mastered a skill [15] in order to provide her with the next adequate learning material. In this situation, BKT is not used only to evaluate the probability that the student will give a correct answer at time t. It is also used to check whether the "p_known" value calculated on fitted model parameters has reached the 0.95 threshold [15]. In that case, inferring learning mastery based on fitted parameter values is risky when there is uncertainty on the fitted values. First, there is a risk that different combinations of parameters may yield functionally identical models that explain observations in the same way. This is known as the *identifiability issue*, an important problem that keeps being discussed and solved in the BKT community [2, 7]. A second issue involves the reliability and confidence in the fitted parameter values. In other words, how sure we are of the fitted parameter value that will be used to infer that the learning mastery threshold has been reached. That issue has been of primary importance in recent usage of AFM to perform advanced learning factor analysis in the field [8] or when building tools to tentatively offer guidance for building competency frameworks [9]. For instance, Durand et al. [9] describe a situation where a skill was first fitted as fairly difficult (low β) with fast learning rate (high γ). After a small modification of the training dataset, the same skill was estimated easy (large β) with no learning (small γ). In addition, it is also known from the literature that latent variable models, including skill-based cognitive models such as AFM, are difficult to estimate precisely [18]. In light of these results, it becomes crucial to take a closer look at the uncertainty on model parameters, beyond predictive accuracy. Quantifying the uncertainty on fitted parameter values by estimating their standard error appears necessary in order to increase our ability to make correct, and hopefully useful, inference from fitted models.

The rest of the article is organized as follows. The next section presents related works. Section 3 presents the AFM model, its use for diagnosing learning, and the computation of the standard error on fitted parameter values, using two different techniques. Experimental results on several cognitive models from the PSLC-Datashop [11] are presented in Section 4 and discussed in Section 5. We then summarize the contributions presented in this paper and their impact on future developments.

2. RELATED WORK

A recent and fundamental paper by Philipp et al. [17] investigates the estimation of Standard Errors in cognitive diagnostics models. Clearly identifying the need of assessing the uncertainty of the estimated model parameters using confidence intervals, they presented the theoretical background for estimating parameter standard errors for the G-DINA cognitive diagnostic model [17]. In their explanations, they essentially presented and discussed different ways of computing standard errors by either considering the complete or the incomplete information matrix. In their experiments, they managed to highlight the necessity of considering the complete information matrix rather than using the incomplete one to compute parameters standard error. This result, while interesting, was not the only focus of our interest. The authors detailed two ways of computing both the complete and incomplete information matrix in the context of G-DINA that were of primary relevance for an application to AFM. The first way uses an Outer Product of Gradient (OPG) estimator. This estimator has the advantage to be relatively easy to implement but slightly less precise than the method using the Hessian of the log-likelihood, which has the drawback of being more cumbersome to implement. In our experiments we used the Hessian estimator of the information matrix.

Computation of the standard error of parameter estimates is a classic approach in statistics method and a dense literature details its applications. However, it seems to have drawn a limited interest in the EDM community so far, as we did not find implementation examples in the EDM literature. Nevertheless, a connecting point could be found in the renewed interest on model identifiability issues [2, 7]. Identifiability issues can lead to an information matrix that is ill conditioned and that cannot be inverted. As we will see later, parameter standard error is obtained by inverting the information matrix using OPG or Hessian approaches. If the information matrix cannot be inverted, there is no standard error that can be obtained by these methods. Philip et al. mentioned that such situation can occur in the DINA model [6] whenever a "test does not involve a single-attribute item for each of the K attributes" [17]. This is a result we intuitively implemented in rules when guiding competency framework refinement with AFM [9]. Howeverm this intuitive ruleturns out to be a requirement for standard error estimation. While BKT identifiability conditions are starting to be well documented, we have not been able to find an equivalent for AFM and we hope that the scientific community will address this issue. The main objective of this contribution is to present, illustrate, and discuss the implementation of AFM parameter standard error estimation. To the best of our knowledge, this had not been addressed yet in the literature.

3. THE ADDITIVE FACTOR MODEL

The AFM [3] models the probability that a student i succeeds on an item j by a mixed-effect logistic regression:

$$P(Y_{ij} = 1|\alpha_i, \beta, \gamma) = \text{logit}^{-1}(\alpha_i + \sum_{k=1}^{K} \beta_k q_{jk} + \sum_{k=1}^{K} \gamma_k q_{jk} t_{ik}) \quad (1)$$

where $\text{logit}^{-1}(x) = 1/(1 + e^{-x})$. Parameters α_i, β_k and γ_k represent the proficiency of student i, easiness of skill k and learning rate for skill k, respectively.[1] The Q-matrix $Q = [q_{jk}]$, also known as the Knowledge Component model in the PSLC-Datashop [11], represents the item-to-skills mapping by a binary matrix, as in the following example:

$$\text{Q} = \begin{array}{c} \\ ItemA \\ ItemB \\ ItemC \\ ItemD \end{array} \begin{array}{c} \begin{array}{ccc} Skill.1 & Skill.2 & Skill.3 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array},$$

where items A, B and D evaluate one skill each, and item C evaluates two.

Finally, variable t_{ik} is the number of times student i has practiced skill k, also known as the *opportunity* number. Parameters β and γ are key differentiators for AFM as a cognitive diagnostics model [8]. They model the learning process for each skill, making AFM a powerful and very unique model to finely characterize the acquisition of skills [8]. Learning parameters allow to plot useful *learning curves* detailing learning acquisition.

3.1 Learning curves

Learning curves are an essential tool to improve learning systems. They "give us a measure of the amount of learning that is taking place relative to the system's model" allowing to compare and improve them [14]. Concretely, a learning curve is a "graph that plots performance on a task versus the number of opportunities to practice" [14]. The performance measured can be the time spent assembling an engine component in a production line or as it is often the case in the educational field, the error rate at applying a set of, or individual skills.

Displaying learning curves in multidimensional learning environments can be difficult. Those environments are not necessary built for single skills learning measurement and they usually combine different set of skills evaluated together (multidimensionality). In such situation, we need to "retrofit" the analysis and AFM is the perfect model to do that as it tries to detect each skill specific (additive) contribution towards each item success.

Learning curves when modeling learning performance over time follow a "power law of practice" [16] which states performance over time should increase following a power law. In the Intelligent Tutoring Systems (ITSs) context, we can expect the error rate to drop as a power law over practice opportunities. Comparing ITS or sections of them can be done by considering the steepness of the curve. A steeper curve indicates a faster acquisitions of the skills practiced [14].

Another advantage of using AFM to draw learning curves is that we can compensate for the *attrition bias*. Over time, fewer learners tend to perform the items because many of them have learned the skill and the curves tend to quickly degenerate, impacting the value of slopes and the power law

[1] We refer to β and γ as the *skill* and *learning* parameters in the rest of the article.

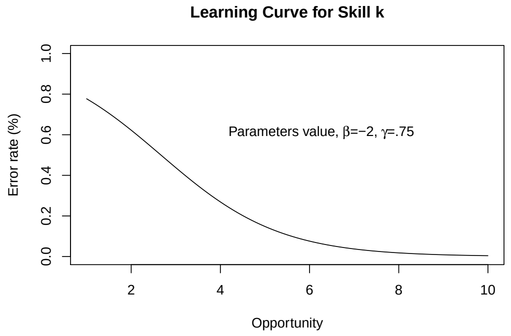


Figure 1: Example of a error curve for a moderately hard skill with a moderately fast learning rate.

fit. A convenient way to produce a learning curve for skill k in AFM is to use Eq. 1 with β_k, γ_k, and a "typical" value of the student proficiency. Using $\alpha_i = 0$ is convenient, and usually roughly corresponds to the average value of the estimated α's. This individual theoretical learning curve for skill k is given by:

$$LC_k(t) = \text{logit}^{-1}(\beta_k + \gamma_k t) = \frac{1}{1 + \exp(-\beta_k - \gamma_k t)}. \quad (2)$$

Typically, we consider *error curves* while talking about learning curves. The error curve is obtained by plotting $EC_k(t) = 1 - LC_k(t)$ as illustrated in Figure 1.

3.2 Computing the Standard Error

We present two methods to estimate the standard errors on parameters. The first one is a classical approach in the statistics literature. It involves the computation of the negative Hessian of the log-likelihood. The second one is inspired by the parametric bootstrap and estimates the standard error by computing empirical standard deviations on the parameters obtained from simulated observation samples.

3.2.1 Negative Hessian of the log-likelihood

Technically, the standard errors of estimated parameters can be retrieved from the covariance matrix of the parameters (eq. 3). More precisely, they are equal to the square root of the diagonal elements in:

$$Cov(\alpha, \beta, \gamma) = \begin{pmatrix} V_\alpha & V_{\alpha,\beta} & V_{\alpha,\gamma} \\ V_{\beta,\alpha} & V_\beta & V_{\beta,\gamma} \\ V_{\gamma,\alpha} & V_{\gamma,\beta} & V_\gamma \end{pmatrix}. \quad (3)$$

However, this covariance matrix is not known and we need to estimate it in order to compute our standard errors. Fortunately, the estimation of covariance matrices have been of interests of statisticians for a long time and several ways have been proposed to solve it. More precisely, it turns out that the covariance matrix is equal to the inverse of the information matrix [17], $Cov(\alpha, \beta, \gamma) = \mathcal{I}(\alpha, \beta, \gamma)^{-1}$. This means we can compute estimators of standard deviation on parameter estimates as long as we can compute and invert the information matrix. At the maximum likelihood, $\mathcal{I}$ is given by the negative Hessian matrix of the log-likelihood:

$$\mathcal{H}_(\mathcal{L}) = \begin{pmatrix} \frac{\partial^2 \mathcal{L}}{\partial^2 \alpha} & \frac{\partial^2 \mathcal{L}}{\partial \alpha \partial \beta} & \frac{\partial^2 \mathcal{L}}{\partial \alpha \partial \gamma} \\ \frac{\partial^2 \mathcal{L}}{\partial \beta \partial \alpha} & \frac{\partial^2 \mathcal{L}}{\partial^2 \beta} & \frac{\partial^2 \mathcal{L}}{\partial \beta \partial \gamma} \\ \frac{\partial^2 \mathcal{L}}{\partial \gamma \partial \alpha} & \frac{\partial^2 \mathcal{L}}{\partial \gamma \partial \beta} & \frac{\partial^2 \mathcal{L}}{\partial^2 \gamma} \end{pmatrix} \quad (4)$$

In our implementation of AFM, we use a penalized version of the log-likelihood, as detailed in [8], and adapt Eq. 4 accordingly.

3.2.2 Simulation

Keeping in mind that "a standard error is the standard deviation of the distribution of parameter estimates over multiple samples" [20], we simulate multiple samples from the initial data, estimate parameters on each samples, and calculate the empirical standard deviation on these results:

Algorithm 1: Pseudo-code of the simulated standard error estimation function. Values in square brackets are defaults.

```
Data: Q-matrix $Q$, first attempt observations $O$ and $\alpha$, $\beta$,
      $\gamma$ parameter values
Parameters: Penalization parameter $\lambda$ [1], number of
            simulations $n$ [1000]
Result: std($\alpha$), std($\beta$), std($\gamma$)
Compute $P(Y_{ij} = 1|\alpha_i, \beta, \gamma)$ according to Eq. 1 for each
  first attempt observation $O_{ij}$;
repeat
    Create $R$, a matrix of P size with random values
      between 0 and 1;
    Create $O'$ a matrix equal to $O$;
    for first attempt observation $O_{ij}$ do
        if $R_{ij} > P(Y_{ij})$ then
            $O'_{ij} \longleftarrow 0$;
    Estimate $\alpha$, $\beta$, $\gamma$ for each simulation iteration with
      respect to $Q$ and $O'$;
until n simulation iterations;
std($\alpha$) ⟵ Standard deviation of $n$ simulation estimated $\alpha$;
std($\beta$) ⟵ Standard deviation of $n$ simulation estimated $\beta$;
std($\gamma$) ⟵ Standard deviation of $n$ simulation estimated $\gamma$;
```

This simulation approach aimed at providing us with an alternative method to validate the Hessian's detailed in previous section but also to provide us with an alternative should inverting the Hessian matrix would be impossible or too cumbersome to implement outside of our experimental environment. The simulation takes as input a Q-matrix and performance observations. It fits the AFM parameters before computing a prediction for each observation. If the prediction is below a random value uniformly distributed between 0 and 1 then the observation is changed to a failure. Then we iterate again by computing new values of AFM parameters on the new observations dataset, computing the predictions and creating another observations sample. The pseudo-code of this simulation process is presented in Algorithm 1.

We also tried another estimation method using a Jackknife approach (iterative leave-one-out on students) that provided us with overly optimistic values. Standard errors were clearly underestimated in the PSLC dataset we experimented.

Table 1: Overall predictive quality of KC models as computed by PSLC-Datashop

Model Name	KCs	#Obs.	AIC	BIC	RMSE
Arith0	18	5,104	4,948	5,569	.397095
Context	12	5,104	5,030	5,573	.399431
Original	15	5,104	5,180	5,762	.407192

4. EXPERIMENTS

4.1 Dataset

In our experiments, we used the "Geometry Area (1996-97)" dataset from DataShop [11]. It contains 6778 observations of the performance of 59 students completing 139 unique items from the "area unit" of the Geometry Cognitive Tutor course (school year 1996-1997). This is a classic Datashop collection, associated with many prior publications [3, 4, 12, 13]. We selected three Knowledge components (KCs) models to run our experiments:

- hLFASearchAICWholeModel3arith0 (Arith0 henceforth);
- hLFASearchModel1-context (Context hereafter);
- Original.

They were selected for their reasonable numbers of skills and observations but also because they have distinctive goodness of fit metrics allowing to differentiate their predictive qualities. Characteristics of these KC models, as reported in Datashop are presented in Table 1. This suggests that the best predictive model would be Arith0, followed by Context and Original. The number of skills (KCs) do not seem to correlate with the goodness of fit for these models.

4.2 Method

Our implementations are done using Matlab and Octave.[2] The AFM estimation used in previous work[8, 9], was extended with the developments described above. The Hessian of the log-likelihood was computed using an off the shelf numerical method using a central difference approximation.[3] This has the advantage of requiring no calculus for computing second derivatives, but has the disadvantage of being notably slower than direct Hessian computation. The full Hessian computation takes around three hours on a regular laptop, for each of the KC models. The simulation-based estimates were obtained using a Go language implementation of AFM parameter estimation. It takes less than 15 minutes in Go to compute 1000 simulation iterations.

4.3 Results

Table 5 shows the estimated values and standard errors for learning parameters β and γ for KC models Arith0, Context and Original. At first glance, we can see that none of the parameters take large values compared to the others. This suggests that the KC models are of excellent quality. Overall inter-model differences in parameter values and standard errors are also relatively small.

Table 2: Mean parameter values

KC Model	Mean parameter values		
	α	β	γ
Arith0	0(.639)	.367(1.261)	.199(.269)
Context	0(.647)	.205(1.323)	.185(.327)
Original	0(.624)	.308(.877)	.147(.127)

Table 3: Mean standard Errors computed with the Hessian

KC Model	Mean standard errors		
	α	β	γ
Arith0	.366(.149)	.349(.137)	.083(.075)
Context	.364(.149)	.320(.175)	.073(.093)
Original	.361(.149)	.284(.073)	.051(.038)

Mean parameter values (across models) in Table 2 show that all models share the same (at .001 precision) mean and almost identical standard deviations of α. This suggest that changing the KC model had a limited impact on students' proficiencies. In other words, students proficiencies remain consistently estimated from one model to another. It seems unlikely that a student proficiency would drastically change from one model to another. Interestingly the mean values of γ are higher in the better models but the standard deviation also increases suggesting higher values with more variance. If we look at the mean standard errors in Table 3, we notice that it is very similar between models for α, suggesting again a limited impact of the KC models on students proficiencies. However the values obtained for learning parameters are very interesting as the mean standard errors increase with the predictive quality of the models. One would have excepted the opposite to happen as Arith0 is expected to have a better fit of the observations than Original. In addition, standard deviations on the errors are also higher for Arith0 than Original. One assumption could be that Arith0 managed to get few better curves with more bad ones and less average good ones. More investigation would be necessary to clarify this point.

5. DISCUSSION

5.1 Model goodness of fit

The dataset used in this experiment is very adapted to conduct learning factor analysis and it is advertised as a good one to showcase PSLC-Datashop features. Consequently the discrepancy obtained between goodness of fit and mean standard error may not generalize to other situations. In addition, we have little knowledge of the intention that led to the design of these KC models. Those cautionary considerations made, we still have been able to characterize a situation were an overall better model does not necessarily lead to a a more reliable KC model. This is an interesting result, for instance, if we want to automatically refine models as in learning factor analysis as it would imply to not only look at model goodness of fit but also KC model goodness of fit. Standard errors can also inform us on the problematic skills to modify as it allow us to get a better grasp on the reliability of learning parameters for each skill.

5.2 Learning detection

[2] Octave/Matlab implementations are available on request.

[3] Octave `Optim` package, `numhessian` function.

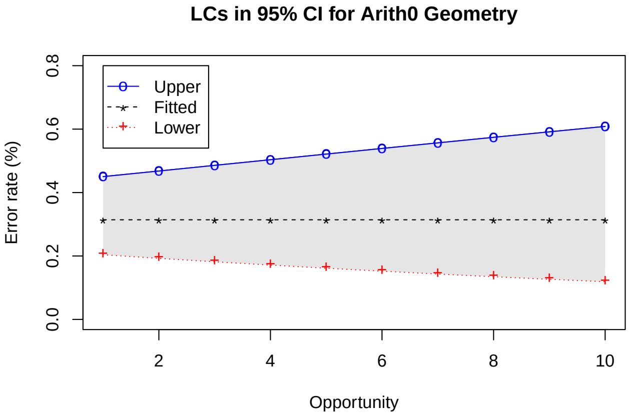


Figure 2: A skill with a flat curve suggesting limited learning for most values in the 95% confidence interval

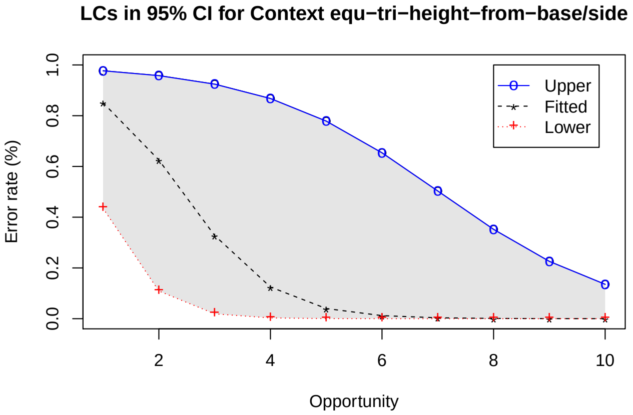


Figure 3: A skill with a steep curve clearly showing learning for all values in its 95% confidence interval

Standard errors allow us to compute confidence intervals on parameters and learning curves. Figures 2 and 3 plot learning curves for a skill with low difficulty and no learning (Fig. 2) and a difficult skill with fast learning rate (Fig. 3). In both cases, the "Fitted" learning curve uses fitted learning parameters, the "Upper" curve is obtained using the parameters at the lower end of the confidence interval (1.96 × StdErr below fitted values), and the "Lower" curve uses parameters are the top of the C.I. (1.96 × StdErr above fitted). The Upper and Lower curves provide us with the extreme slopes that the learning curve can take in a 95% confidence interval, and show the range of difficulty the skill can take while still remaining in the confidence interval.

Some values taken by these curves are not possible in practice. For instance in Figure 2, the Upper curve is impossible under AFM parameter fitting constraints, as γ is constrained to be positive. On the other hand, the Lower curve can be observed and shows limited learning. In this configuration of learning parameters, stating no learning after looking only at the Fitted learning curve could be an overstatement even though it is very likely that no learning is occurring. However as Murray et al. [15] showed, flat aggregated curves showing no learning could, in fact, hide the learning occurring for sub-group of students. In their study of an algebra curriculum containing performance data of 15,414 students on 881 skills, they discovered that around 16% of skills were misidentified as showing no learning. Standard error computation gives another reason why we should be cautious when claiming no learning. But can standard errors help us claim learning? The skill in Figure 3 answers this question. We can see that all the difficulties and slopes that can be taken in the 95% confidence interval leads to conclude that this skill is learned. In conclusion to this subsection, considering fitted parameter standard errors is important to confirm that learning is occurring but not necessarily the opposite.

Table 4: RMSE and r^2 computed between the Hessian and the simulation standard errors

KC Model	RMSE			r^2		
	α	β	γ	α	β	γ
Arith0	.052	.050	.022	.906	.963	.987
Context	.053	.061	.020	.890	.900	.973
Original	.047	.026	.004	.917	.947	.992

5.3 Simulation and Hessian methods

Table 5 shows that standard errors computed from the log-likelihood Hessian and by simulation are very close. This means that our method can potentially provide an estimate of the standard errors when the Hessian is hard to compute or invert. This also confirms the validity of our simulation results. Table 4 shows the Root Mean Square Error (RMSE) and correlation (r^2) between simulation estimates and the standard errors over all parameters of each KC Model. Although not insignificant, the difference between the two methods is sufficiently small, and the value of r^2 large enough, to consider that simulation results provide good estimates of the standard errors on parameters.

6. CONCLUSION AND FUTURE WORK

Estimating the reliability of parameter estimates is a crucial aspect of model inference. We showed how to compute standard errors on AFM model parameters, and applied the proposed methods to public datasets from the PSLC Datashop. This yields several observations.

First, the more accurate model is not always the one with the better KC model: parameter validity and predictive ability are different. That confusion is not new however and allowed progress in cognitive psychology in the first half of the nineteenth century before the community realized it failed to "provide a strong foundation for deducing likely relationships among variables, and hence for the development of generative theory"[19].

Second, standard errors, and the associated confidence intervals, provide precious insight into learning. However, characterizing the absence of learning is more complicated, especially when γ is less reliable.

Finally, standard errors on parameters can be easily estimated by the simulation method we describe. This can be

Table 5: Estimated parameters and standard errors for several PSLC models.

Model	Skill	β	StErrβ	Simul.	γ	StErr γ	Simul.
Arith0	Geometry*parallelogram-area	1.939	0.233	0.224	0.028	0.016	0.016
Arith0	Geometry*parallelogram-area*Textbk_New_Decomp...	2.540	0.617	0.659	0.180	0.149	0.192
Arith0	Geometry*Textbk_New_Decompose-circle-area	1.136	0.374	0.399	0.183	0.093	0.111
Arith0	arithmetic	1.992	0.272	0.250	0.027	0.023	0.022
Arith0	Geometry	0.781	0.260	0.197	0.000	0.036	0.021
Arith0	Geometry*decomp-trap*trapezoid-area	-0.624	0.200	0.202	0.092	0.017	0.017
Arith0	Geometry*ALT:TRIANGLE-AREA	1.501	0.341	0.260	0.000	0.056	0.035
Arith0	Geometry*ALT:TRIANGLE-AREA-PART	0.204	0.400	0.416	0.230	0.124	0.132
Arith0	Geometry*compose-by-multiplication	-0.675	0.390	0.400	0.267	0.121	0.126
Arith0	Geometry*pentagon-area	-0.550	0.199	0.200	0.110	0.015	0.016
Arith0	Geometry*ALT:CIRCLE-AREA-INDIRECT	-0.268	0.305	0.306	0.312	0.066	0.071
Arith0	Geometry*Textbk_New_Decompose-circle-area*circle...	0.871	0.255	0.258	0.073	0.030	0.031
Arith0	Geometry*ALT:CIRCLE-AREA	0.973	0.280	0.281	0.124	0.039	0.042
Arith0	Geometry*circle-area	-0.393	0.348	0.342	0.171	0.089	0.093
Arith0	Geometry*circle-diam-from-subgoal	0.126	0.275	0.268	0.071	0.045	0.043
Arith0	Geometry*equi-tri-height?	-2.986	0.714	0.888	1.232	0.310	0.385
Arith0	Geometry*decomp-trap	-0.555	0.304	0.304	0.146	0.057	0.060
Arith0	compose-subtract	0.588	0.524	0.540	0.329	0.200	0.222
Context	parallelogram-area	2.105	0.234	0.227	0.019	0.012	0.012
Context	context	0.105	0.168	0.117	0.000	0.005	0.002
Context	Geometry	0.873	0.168	0.171	0.016	0.005	0.006
Context	Subtract-rectangles	2.475	0.571	0.398	0.000	0.137	0.091
Context	decomp-trap	-0.529	0.181	0.184	0.060	0.012	0.012
Context	compose-by-multiplication	0.284	0.248	0.245	0.114	0.023	0.023
Context	pentagon-area	-0.552	0.199	0.197	0.110	0.015	0.016
Context	circle-area	0.393	0.212	0.217	0.106	0.019	0.020
Context	radius-from-area	-0.427	0.351	0.347	0.165	0.089	0.091
Context	radius-from-circumference	0.134	0.275	0.269	0.067	0.045	0.044
Context	equ-tri-height-from-base/side	-2.972	0.713	0.819	1.230	0.310	0.354
Context	Subtract	0.576	0.523	0.554	0.336	0.200	0.227
Original	ALT:PARALLELOGRAM-AREA	2.326	0.250	0.197	0.011	0.016	0.013
Original	ALT:PARALLELOGRAM-SIDE	1.054	0.494	0.473	0.345	0.152	0.157
Original	ALT:COMPOSE-BY-ADDITION	1.035	0.191	0.135	0.000	0.012	0.008
Original	ALT:TRAPEZOID-AREA	-0.860	0.344	0.340	0.344	0.092	0.094
Original	ALT:TRAPEZOID-HEIGHT	-0.800	0.329	0.340	0.243	0.079	0.083
Original	ALT:TRAPEZOID-BASE	-0.498	0.334	0.334	0.233	0.084	0.085
Original	ALT:TRIANGLE-AREA	0.964	0.249	0.237	0.042	0.028	0.027
Original	ALT:TRIANGLE-SIDE	0.122	0.297	0.245	0.037	0.056	0.044
Original	ALT:COMPOSE-BY-MULTIPLICATION	0.393	0.231	0.221	0.113	0.022	0.023
Original	ALT:PENTAGON-AREA	-1.000	0.334	0.327	0.392	0.081	0.083
Original	ALT:PENTAGON-SIDE	-0.413	0.235	0.226	0.151	0.028	0.029
Original	ALT:CIRCLE-RADIUS	0.360	0.234	0.210	0.046	0.027	0.026
Original	ALT:CIRCLE-AREA	0.473	0.209	0.197	0.104	0.019	0.020
Original	ALT:CIRCLE-CIRCUMFERENCE	0.876	0.268	0.251	0.073	0.037	0.037
Original	ALT:CIRCLE-DIAMETER	0.593	0.258	0.252	0.074	0.034	0.036

convenient when the Hessian of the log-likelihood is not easily calculated or inverted.

Our work also raised significant questions. For instance, the identifiability of the AFM model needs to be addressed, as it is likely that AFM could, like DINA be in trouble on a dataset that “does not involve a single-attribute item for each of the K attributes” [17].

7. ACKNOWLEDGMENTS

We wish to thank Mimi McLaughlin and Cindy Tipper for help and clarification with the datasets in Datashop.

8. REFERENCES

[1] KDD cup 2010, student performance evaluation. `http://www.kdd.org/kdd-cup/view/kdd-cup-2010-student-performance-evaluation/Data`[Accessed: 2018-03-07].

[2] J. E. Beck and K.-M. Chang. Identifiability: A fundamental problem of student modeling. In C. Conati, K. McCoy, and G. Paliouras, editors, *User Modeling 2007*, pages 137–146, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[3] H. Cen, K. Koedinger, and B. Junker. Learning factors analysis – A general method for cognitive model evaluation and improvement. In M. Ikeda,

K. D. Ashley, and T.-W. Chan, editors, *Intelligent Tutoring Systems: 8th International Conference, ITS 2006, Jhongli, Taiwan, June 26-30, 2006. Proceedings*, pages 164–175, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[4] H. Cen, K. Koedinger, and B. Junker. Is overpractice necessary? — Improving learning efficiency with the cognitive tutor through educational data mining. In R. Luckin, K. R. Koedinger, and J. Greer, editors, *Proceeding of the 2007 Conference on Artificial intelligence in Education: Building Technology Rich Learning Contexts that Work*, number 158 in Frontiers in Artificial Intelligence and Applications, pages 511–518, Amsterdam, 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, 1994.

[6] J. De la Torre. DINA model and parameter estimation: A didactic. *Journal of Educational and Behavioral Statistics*, 34(1):115–130, Mar. 2009.

[7] S. Doroudi and E. Brunskill. The misidentified identifiability problem in bayesian knowledge tracing. In *Proceedings of the 10th International Conference on Educational Data Mining.*, pages 143–149. EDM, 2017.

[8] G. Durand, C. Goutte, N. Belacel, Y. Bouslimani, and S. Léger. Review, computation and application of the additive factor model (AFM). Tech. Report 23002483, National Research Council Canada, 2017.

[9] G. Durand, C. Goutte, N. Belacel, Y. Bouslimani, and S. Léger. A diagnostic tool for competency-based program engineering. In *Proceedings of the Eight International Learning Analytics &#38; Knowledge Conference*, LAK '18, New York, NY, USA, 2018. ACM.

[10] J. Gonzalez-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 *The 8th International Conference on Educational Data Mining*, 2015.

[11] K. Koedinger, R. 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. Baker, editors, *Handbook of Educational Data Mining*. CRC Press, 2010.

[12] K. R. Koedinger, K. Cunningham, A. Skogsholm, and B. Leber. An open repository and analysis tools for fine-grained, longitudinal learner data. In *EDM*, pages 157–166. www.educationaldatamining.org, 2008.

[13] K. R. Koedinger, E. A. McLaughlin, and J. C. Stamper. Automated student model improvement. In *EDM*, pages 17–24. www.educationaldatamining.org, 2012.

[14] 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, Aug 2011.

[15] R. C. Murray, S. Ritter, T. Nixon, R. Schwiebert, R. G. M. Hausmann, B. Towle, S. E. Fancsali, and A. Vuong. Revealing the learning in learning curves. In H. C. Lane, K. Yacef, J. Mostow, and P. Pavlik, editors, *Artificial Intelligence in Education*, pages 473–482, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[16] A. Newell and P. S. Rosenbloom. Mechanisms of skill acquisition and the law of practice. *Cognitive skills and their acquisition*, 1:1–55, 1981.

[17] M. Philipp, C. Strobl, J. de la Torre, and A. Zeileis. On the estimation of standard errors in cognitive diagnosis models. *Journal of Educational and Behavioral Statistics*, 2017.

[18] I. Ropovik. A cautionary note on testing latent variable models. *Frontiers in Psychology*, 6:1715, 2015.

[19] M. E. Strauss and G. T. Smith. Construct validity: Advances in theory and methodology. *Annual Review of Clinical Psychology*, 5(1):1–25, 2009.

[20] T. Verguts and G. Storms. Assessing the informational value of parameter estimates in cognitive models. *Behavior Research Methods, Instruments, & Computers*, 36(1):1–10, Feb 2004.

Exploring Collaboration Using Motion Sensors and Multi-Modal Learning Analytics

Joseph M. Reilly
Harvard Graduate School of Education
13 Appian Way
Cambridge, MA 02138
1-412-956-6116
josephreilly@g.harvard.edu

Milan Ravenell
Harvard University
Massachusetts Hall
Cambridge, MA 02138
1-617-495-1000
mravenell@college.harvard.edu

Bertrand Schneider
Harvard Graduate School of Education
13 Appian Way
Cambridge, MA 02138
1-617-496-2094
bertrand_schneider@g.harvard.edu

ABSTRACT

In this paper, we describe the analysis of multimodal data collected on small collaborative learning groups. In a previous study [1], we asked pairs (N=84) with no programming experience to program a robot to solve a series of mazes. The quality of the dyad's collaboration was evaluated, and two interventions were implemented to support collaborative learning. In the current study, we present the analysis of Kinect^{TM} and speech data gathered on dyads during the programming task. We first show how certain movements and patterns of gestures correlate positively with collaboration and learning gains. We next use clustering algorithms to find prototypical body positions of participants and relate amount of time spent in certain postures with learning gains as in Schneider & Blikstein's work [2]. Finally, we examine measures of proxemics and physical orientation within the dyads to explore how to detect good collaboration. We discuss the relevance of these results to designing and assessing collaborative small group activities and outline future work related to other collected sensor data.

Keywords

Multi-modal learning analytics, physical synchrony, computational thinking, collaboration

1. INTRODUCTION

Collaboration is increasingly listed as a common factor in many frameworks of 21st Century Skills that highlight how classrooms and workplaces will differ from their traditional models due to deluges of digital data from information and communications technologies [3]. Likewise, computational thinking has been deemed an essential set of skills and attitudes that are now central to all science, technology, engineering, and mathematical (STEM) disciplines as well as computer science [4]. The ability to rapidly assess and evaluate collaborative computational thinking tasks can facilitate instruction that aligns with these important aspects of modern learning environments.

Multi-modal learning analytics utilizing multiple sensor technologies and machine learning techniques can offer insights into student learning in complex, open-ended scenarios such as computer programming, robotics, and problem-based learning [5]. These methods allow researchers and educators to conduct quantitative research without necessarily losing the richness of open-ended, constructionist activities [6]. These techniques are intended to be scalable and help implement better instruction by generating formative feedback, visualizing performance, and increasing the salience of important information for instructors.

This paper focuses on measuring the quality of collaboration by analyzing participant movement and correlating a variety of measures with task performance and a coding scheme for assessing collaboration quality in dyads. We first summarize relevant literature on collaborative problem solving and the importance of gesturing in collaboration. Next, we explain the design and methods of the study where our data originated. Finally, we report our current findings and describe future work for our research.

2. LITERATURE REVIEW

2.1 Collaborative Problem Solving

Researchers in computer-supported collaborative learning (CSCL) have long studied how small groups collaborate and co-construct knowledge [7]. The joint problem space that collaboration entails requires active social negotiation of the current problem, what can be done to solve the problem, and the goals of the task [8]. By studying how collaboration proceeds at a fine-grained level, researchers can assess the quality of this collaboration and see what measurable markers denote high quality collaboration. Examples of such dimensions include synchrony of physical actions and eye gaze [2, 9], physical reactions of participants to the actions of others [10], and gestures made during activities [11].

2.2 Gestures and Movement in Collaboration

Emerging literature from multi-modal learning analytics has explored the roles of gesture, posture, and gross motor movement in collaborative, co-located activities. For example, facial expressions and gestures related to the face predict engagement and frustration, while facial expression and body posture have been shown to predict learning [12]. Bimanual coordination has been shown to be predictive expertise, where experts use both hands in a construction task more equally than novices [13]. Researchers have also been able to predict agreement between participants with a 75% accuracy using motion sensors and audio data streams [21]. Automatically detected measures of non-verbal synchrony (computed from Kinect data) have been found to predict creativity in dyads [22]. Finally, interactive tabletops have been a fruitful area of research for studying collaborative learning groups; motion sensors and microphones have been used to capture students' interactions and provide feedback to teachers about the status of the group [23].

Even if meanings of gestures cannot be automatically deduced from sensor data, the amounts of gesticulation can be calculated and used to augment analysis of learning [14]. While expert coders in a qualitative study can extract context-dependent meaning from a wide variety of gestures [15], quantitative work can utilize unsupervised machine learning methods to cluster student postures and movement patterns automatically to gain a coarse-grained sense of how students are transitioning between states in an activity and how those state transitions relate to learning gains and collaboration measures [2].

This paper builds upon this emerging literature to look at students' micro-behaviors during their learning process (e.g., [20]). More specifically, we explore how unsupervised machine learning algorithms can find prototypical states from dyads of students when learning to program a robot.

3. The Study

3.1 Participants

Forty-two dyads completed the study (N = 84) and forty groups were used in the final data set (each researcher's first session was removed to improve overall fidelity.) Participants were drawn from an existing study pool at a university in the northeastern United States. 62.2% of participants reported being students, with ages ranging from 19 to 51 years old with a mean age of 26.7 years. 60% of participants identified as female.

3.2 Design & Procedure

Employing a two-by-two between-subjects design, dyads were randomly assigned to one of four conditions: Condition #1 received neither intervention, Condition #2 received solely a visualization intervention, Condition #3 received solely an informational intervention, and Condition #4 received both interventions. The informational intervention was delivered verbally by the researcher and consisted of several research findings relevant to collaborative tasks such as equity of speech time predicting the overall quality of a collaboration. The visualization intervention utilized speech data from the motion sensor to visualize the relative proportion of speech coming from each participant over the prior 30 seconds of the activity. Each participant was represented by a color on their side of the tablet, and the screen would fill with more or less of their color to reflect their contribution (see Figure 1, right).

After signing informed consent paperwork, participants were fitted with sensors described in 3.4. Participants were shown a tutorial video illustrating the basics of writing a simple program in Tinker, a block-based programming language. Participants then had five minutes to write code to move a simple robot across a line on the table roughly two feet in front of it. The robot consisted of a microcontroller, two DC motors with wheels, and proximity sensors mounted on the front, right, and left (see Figure 1, left).

Following the tutorial activity, dyads were shows a second tutorial video that highlighted more advanced features of Tinker such as using provided pre-written functions to turn the robot, checking the values of the proximity sensors, and using conditional statements. A hard copy of a reference sheet that summarized the contents of the video was provided following this. Dyads then had 30 minutes to write code to allow the robot to solve a series of increasingly complex mazes (see Figure 1, center). Once the participants' code successfully guided the robot through a maze twice, a new maze was provided. During the main portion of the activity, a series of predetermined hints were given to dyads at 5-minute intervals regarding common pitfalls researchers identified in pilot testing.

3.3 Dependent Measures

The dyad's collaboration and task behaviors were evaluated during the task by the researcher running that session. Quality of collaboration was assessed on nine scales based on Meier, Spada, and Rummel's work [16]: sustaining mutual understanding, dialogue management, information pooling, reaching consensus, task division, time management, technical coordination, reciprocal interaction, and individual task orientation. Task behaviors evaluated were task performance, task understanding, and improvement over time. Following the activity, researchers coded the quality of the final block-based code each dyad produced to determine how well the code could theoretically guide the robot through a maze of unknown layout.

To assess learning of computational thinking skills, participants individually completed a pre- and post-test with four questions assessing principles of computer science such as using conditional statements, looping, and predicting the output of given code (adapted from [17], [18]). Researchers coded the completeness of answers based on their demonstrations of understanding of computational thinking principles. Along with the post-test,

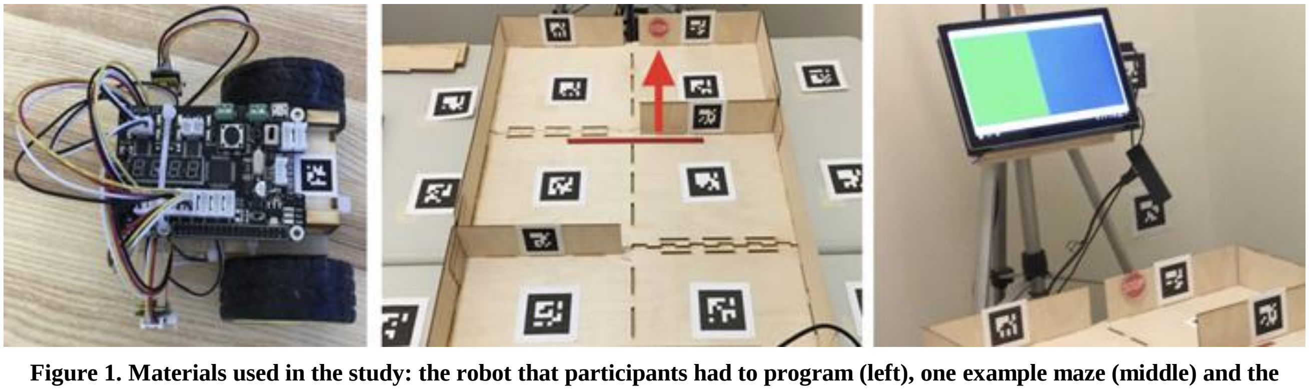


Figure 1. Materials used in the study: the robot that participants had to program (left), one example maze (middle) and the Kinect-based speech visualization (right).

participants completed a self-assessment of the perceived quality of their collaboration with their partner (also adapted from [16]). Participants also filled out demographic information and completed a free response reflection regarding how their thinking changed over time.

3.4 Process Data from Multi-modal Sensors

We used three types of sensors during the study: two mobile eye-trackers captured participants gaze movements at 50Hz; two Empatica wristbands captured physiological signals (e.g., electrodermal activity, heart rate, …) at various rates; and one Kinect sensor captured body postures and facial information. Finally, we also used several cameras and microphones to get an overview of the interaction. The details of the exact sensors used and the types of data collected are available in [1]. In this paper, we focus more closely on the Kinect data.

The Kinect motion sensor collects roughly 100 variables related to a person's body joints and skeleton (24 different points with columns for x, y, z coordinates), their facial expressions, and their amount of speech (Figure 2, top). Typically collected at 30 Hz (30 times per second), this results in roughly 3,000 observations per second or 5.4 million observations per individual during a 30-minute session of our study. When done with dyads, this amount of data doubles.

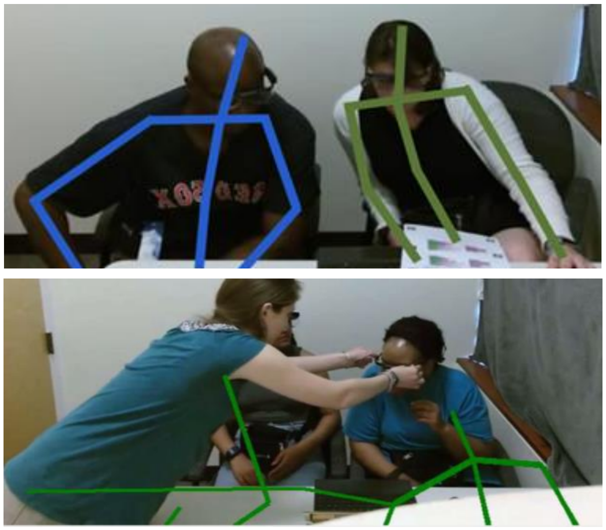


Figure 2. Visual representation of skeletons of participants (top), example of "messy" data caused by researcher entering the frame (bottom).

3.5 Data Preprocessing

Each session's Kinect data contained 8-10 comma separated value (CSV) files as a new file was created every time a participant was lost and then detected again by the motion sensor. After cleaning the data to leave only observations collected during the main portion of the activity, CSV files were assigned to either the left or right participant based on their average spine locations. Experimental design prohibited participants from switching sides during the activity. Additional cleaning was required in instances where researchers briefly entered the frame of the Kinect while the session was underway. This often led to participant wireframes merging or otherwise becoming distorted (Figure 2, bottom). All instances where participant skeletons could not be clearly resolved were removed from our analysis.

After assignment of participant side and cleaning, movement variables were calculated for each of the skeleton points by calculating the difference between the coordinates of a point at one observation and the coordinates of the same point at the next observation. If the skeletal point was occluded from the Kinect sensor (i.e., a hand below the surface of the table) positions of that point were automatically inferred by the sensor but no movement variables were calculated. Joint angles were also calculated for each major joint.

CSV files were combined in two different ways: all were concatenated to give an *individual* level file while left and right participant files were outer joined to create a *dyad* level file. The Kinect data computations for this paper were run in Python 2.7 and analyses of pre-post survey data was done in R 3.4.3 and RStudio 1.1.423.

4. RESULTS

This section summarizes our analyses and results: first, we describe some trends in the dependent measures (4.1). Second, we look at the amount of movement generated by each participant / dyads, and how they correlate with the dependent measures (4.2). Third, we use clustering methods to find prototypical body postures to identify "(un)productive" states (4.3). Finally, we analyze dyadic interactions from the Kinect data (4.4).

4.1 Task Performance and Collaboration

We first briefly describe the main results of the study (also to be reported in [1]). The researcher-coded quality of collaboration differed significantly between the conditions that received the informational intervention (3&4) and those that did not (1&2). Dyads assigned to "explanation" scored 7.1 percentage points higher than those in "no interventions" (p < 0.001). Dyads in "both interventions" scored 4.8 percentage points higher than those in "visualization" (p = 0.03).

Participant individual self-assessments of the quality of their collaboration different significantly from researcher assessment at the dyad level (F = 15.21, p < 0.001) but both are significantly positively correlated (r = 0.43, p = 0.001). Self-reported scores were higher for measures of task division, time management, and reciprocal interaction while being lower for reaching consensus, dialog management, and sustaining mutual understanding.

Participants across all conditions gained an average of 19.8 percentage points on the survey of computational thinking principles (t = 6.18, p < 0.001). Learning gains did not differ significantly by condition, gender, the gender makeup of the group, or level of previous education. Pre-test scores did not differ significantly by condition. The quality of the final block-based code dyads produced was significantly correlated with the number of mazes completed (r = 0.45, p < 0.001), task understanding (r = 0.45, p < 0.001), and improvement over time (r = 0.54, p < 0.001). Significant correlations from these surveys and assessments are summarized in Figure 3.

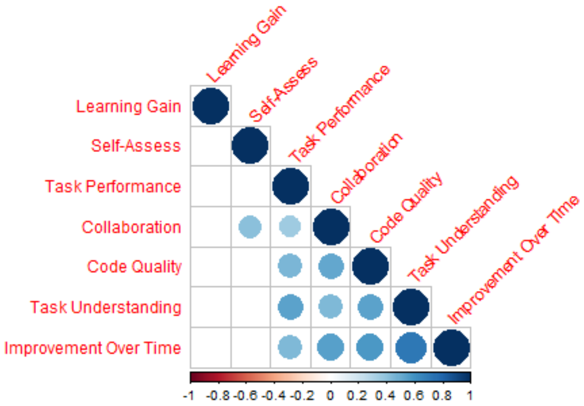


Figure 3. Correlogram of performance metrics and ratings of collaboration. All correlations shown are significant.

4.2 Movement Variables

At the individual level, neither the total movement of any specific joint nor the average movement of those points correlated significantly with any of our collaboration or task performance metrics. Amount of time talking was significantly correlated with total quality of collaboration at the individual level (r = 0.30, p = 0.01) and will be investigated in-depth.

Most of our measures are at the dyad level, so movement variables were aggregated by session rather than participant. Improvement over time was significantly correlated with increased movement of the right elbow (r = 0.47, p = 0.006), right shoulder (r = 0.38, p = 0.029), mid-spine (r = 0.41, p =0.018), and neck (r = 0.38, p = 0.028). Task performance was significantly correlated with right elbow (r = 0.35, p =0.037), right shoulder (r = 0.35, p = 0.035), right hand (r = 0.36, p = 0.027), and mid-spine movement (r = 0.40, p = 0.017). Code quality was significantly correlated with increased movement of the right elbow (r = 0.34, p = 0.025), right shoulder (r = 0.32, p = 0.032), mid-spine (r = 0.31, p =0.017), and neck (r = 0.34, p = 0.024). Overall collaboration more strongly correlated with higher average talk time at the dyad level than the individual level (r = 0.48, p = 0.0008).

Clustering was done on the movement variables to identify patterns of movement that may be relevant to our measures of collaboration and task performance. Due to the unpredictable nature of missing data due to occluded limbs and joints, the 18 movement variables per observation often had one or two missing values. Rather than throw out the entire row, we utilized the K-POD algorithm [19], a variant of k-means clustering that can handle and impute missing data. We generated 2 through 9 clusters and visually inspected the separation of the different centroids. We elected to keep three clusters due to good separation and ease of interpretability.

Groups that spent a higher proportion of their time in the high movement cluster had significantly higher task performance (r = 0.31, p = 0.049) and improvement over time (r = 0.44, p = 0.009). Our overall rating of collaboration did not significantly correlate with time spent in this cluster (p = 0.052) but ratings of reaching consensus and dialogue management did differ significantly (r = 0.34, p = 0.04; r = 0.40, p = 0.02). Individuals overall spent roughly 13% of their time in high movement states with the remainder of their time evenly split between medium and low movement states.

4.3 Angle Variables

In this section, we replicate Schneider & Blikstein (2015)'s approach for identifying prototypical body postures using joint angle. Joint angles were calculated for 11 upper body joints for all observations. Due to having much less missing data for joint angles versus movement variables, k-means clustering was used to generate visualizations of prototypical postures participants held during the course of the activity. As with our prior clustering, 2 through 9 clusters were fit with our model and we chose three clusters due to the interpretability of the resulting visualizations.

As seen in Figure 4, the three postures are distinct in hand placement, symmetry, and arm position. The first posture (left) can be thought of as "planning" where both hands are close together and the participant is leaning forward. This is generally the default posture for someone looking at a computer screen. Dyads spent a large amount of their time looking over their code and the various options available to them.

The second posture (Figure 4, center) we refer to as a "tinkering" state where the robot is being directly manipulated. In this state, participants are generally standing or leaning up out of their chairs to test different scenarios the robot might encounter and what sensor values those scenarios generate. Participants also had to manually reset their robot to the starting position after each attempt to solve a maze.

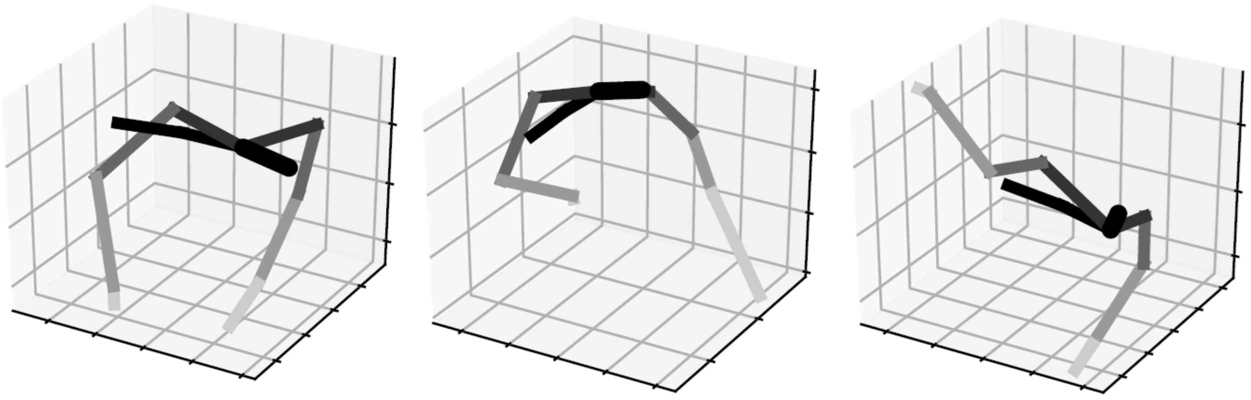


Figure 4. Three prototypical postures participants assumed during the study.

The final state (Figure 4, right) comes from a design decision made in the study. The small robot was tethered to the participant laptop via a USB cord for power and to upload new code, so each time the robot was in motion one participant had to hold the USB cord high enough to avoid it getting tangled in the maze. The prototypical posture shows this clearly. We refer to this posture as “iterating” as it is only observed when running code in an attempt to solve the maze. Examples of the “planning” and “iterating” postures can be seen in Figure 5.

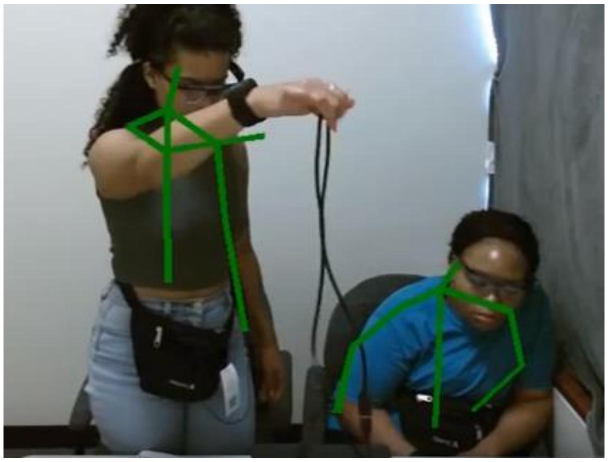
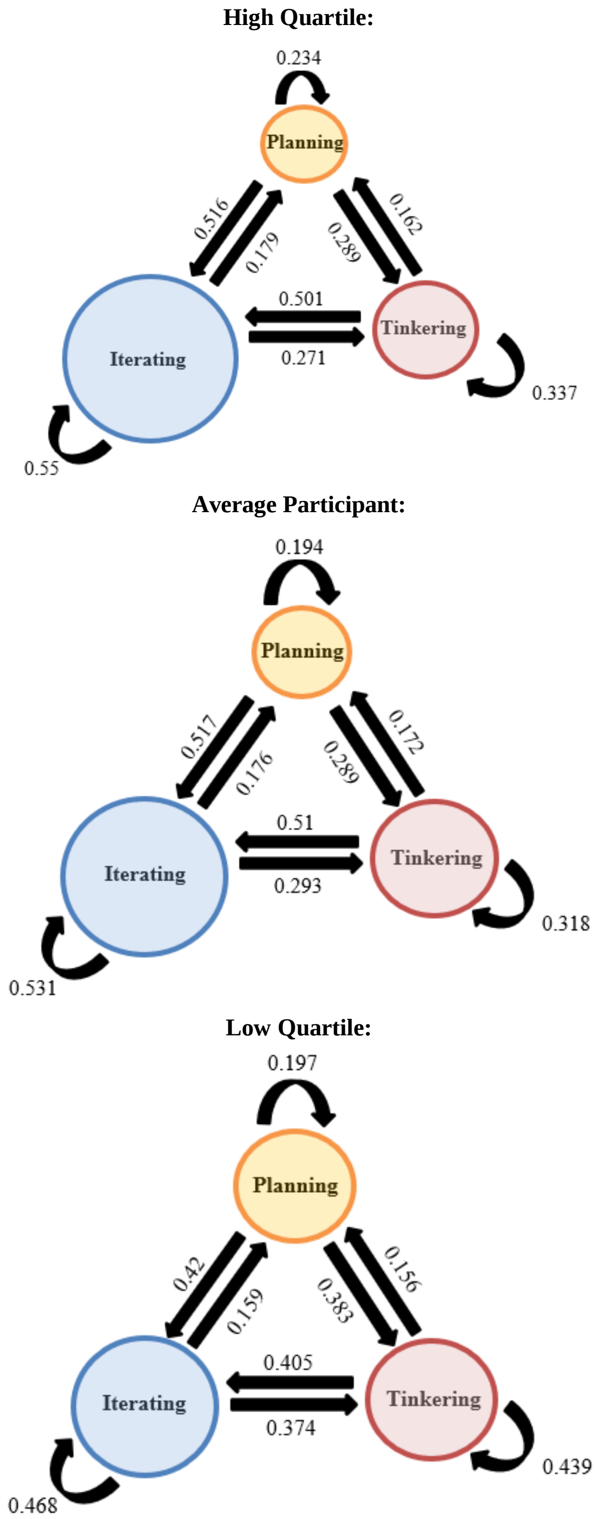


Figure 5. Examples of “iterating” posture (holding wire) and “planning” (seated participant).

As with the movement variables, proportion of time spent in each posture was aggregated for each participant. Increased proportions of time spent in the “iterating” posture significantly correlated with task performance (r = 0.28, p = 0.002), code quality (r = 0.24, p = 0.005), task understanding (r = 0.24, p = 0.02) and improvement over time (r = 0.20, p = 0.02). Proportion of time spent in the “tinkering” posture, however, negatively correlated with the same four metrics: task performance (r = -0.31, p = 0.0004), code quality (r = -0.23, p = 0.008), task understanding (r = -0.27, p = 0.003) and improvement over time (r = -0.27, p = 0.003).

To analyze the probabilities of state transitions taking place between these prototypical postures, a Markov model was constructed to visualize the probabilities of different state transitions occurring (Figure 6). The size of the circles represents the relative amount of time spent in each state and the labels of the arrows indicate the probability of different transitions occurring. The most likely transitions for the average participant (Figure 6, center) all involve the “iterating” state, either staying in it or moving from the other states to it. The least likely transitions involve moving from “iterating” or “tinkering” back to the “planning” state.

Markov models for individuals in the highest performing and lowest performing quartiles (according to their task performance) were generated to explore how state transitions may vary by outcome. High performing individuals (Figure 6, top) were 13% more likely to transition back from “iterating” to “planning” and 38% more likely to transition from “tinkering” to “planning” versus their low performing peers (Figure 6, bottom). High performing individuals spent 12% less time in the “tinkering” state versus low performers, using this time to run more iterations of their code versus adjusting the robot itself.



High Quartile:

Average Participant:

Low Quartile:

Figure 6. Markov state transition models.

4.4 Dyad Interactions

A proximity measure was calculated based on spine positions to determine how closely participants were seated next to each other, a leaning measure determined if participants were leaning towards each other or away from each other, a facing measure based on participant shoulders determined how much participant bodies were facing each other, and bimanual coordination was calculated for each participant to see how evenly they used both of their hands during the activity. While bimanual coordination is calculated at the individual level, the dyad analysis explores whether synchrony in bimanual coordination correlates with our outcome measures.

Performance on the task correlates positively with dyads leaning towards each other ($r = 0.34$, $p = 0.030$). Increased bimanual coordination of the right participant correlates with task understanding ($r = 0.34$, $p = 0.018$) but synchrony of coordination does not seem significant. Due to the setup of the room where the study was conducted, the mouse of the participant laptop was placed on the right side and may have led the right participant to use the laptop more. This may have had an uneven influence on the impact of their bimanual coordination.

Alignment and proximity are strongly correlated ($r = 0.83$, $p < 0.001$) in our dyads but neither measure significantly correlates with our task performance measures. While proximity was not correlated with our overall measure of collaboration, participants being closer together is significantly correlated with information pooling ($r = 0.35$, $p = 0.026$).

5. DISCUSSION

This paper provides some preliminary and promising results describing the relationship between students' body postures / movements and their quality of collaboration, task performance and learning gains. We found predictors for those dependent measures in a naturalistic, open-ended task that routinely takes place in makerspaces and engineering courses. While there are limitations to this work, our contribution paves the way to rich multimodal analyses of students' collaboration. It also unlocks new opportunities to design innovative interventions to support social interactions in small groups (e.g., by providing visual representations of students' behavior to support self-reflection) and classroom orchestration (e.g., by providing teachers with real-time dashboards of the class).

The significant correlations found between average movement of points along the upper right side of participants' bodies with outcome measures indicates the importance of gesturing and physical movement when communicating ideas. Qualitative coding of exemplar videos may detect specific gestures or movements used more frequently by high performing groups, but these movement variables offer a quick way to potentially predict how well participants will do in an activity. While we do not make any causal claims regarding increasing movement to increase performance, future interventions could target visualizing gesture and movement data for dyads as they work instead of verbal contribution.

The clusters generated by our joint angle data reveal interesting patterns in participant behavior. While time spent iterating has been shown here to correlate with better performance, dyads may benefit from more cycling through the three states to mimic ideal cycles of cognition [20]. While iterating and testing their code is certainly important, participants must be able to process what went wrong and try to fix it before attempting to test their code again. In several sessions, participants kept running their code over and over in hopes that the robot would perform better the next time. Even though they had the code in front of them to manipulate, some novices may have lacked the computational thinking knowledge to transfer errors they saw the robot making to errors in their code.

6. LIMITATIONS

We do not have data on the handedness of our participants, but an open question is whether the mouse placement on the right side of the shared laptop inadvertently lead the right participant to assume a leadership role with the laptop. The uneven importance of bimanual coordination for the right participant is an indication the physical setup of the room may have impacted the study in unintended ways. Analyzing the recordings of sessions and identifying leader behavior or who is assuming driver / passenger roles is an additional avenue for future work.

Some of our posture results are fairly idiosyncratic to our study due to the USB cord attached to the robot, making generalization of findings difficult.

As described in Section 3.5, the Kinect sensor generated a wide variety of malformed skeletons that led to a lengthy and imprecise period of manual cleaning prior to analysis. Experimental design must be conscious of the limitations of the sensors and ensure that as little noise as possible be added to the data.

7. FUTURE WORK

We plan to further identify productive micro-behaviors from the Kinect data to gain additional insights in the ways that dyads synchronized their actions. Future work with regards to prototypical postures would also explore both participants in a dyad at once, clustering on both joint angles simultaneously. This may reveal combinations of postures that are informative and could extend our exploration of physical synchrony within dyads. The differences between dyads in different conditions will also be a main focus of analysis moving forward.

It should be noted that this paper only describes one aspect of a positive collaboration. In future work, we plan to extend this line of work to attentional alignment (also referred to as joint visual attention [24]) using the eye-tracking data, verbal coherence [25] using transcripts, physiological synchronization [26] using the Empatica data, and ultimately combine those modalities together. This will provide us with a richer and more comprehensive view of students' collaboration and potentially feed machine learning algorithms to make predictions about the status of a group using multimodal streams of data.

Future work will also revisit our coding of collaboration to improve inter-rater reliability (currently Cronbach's alpha = 0.65, 75% agreement). For our movement clustering, several correlations with collaboration measures were close to being significant but may have been hindered due to less-than-ideal reliability of our initial coding. Additionally, patterns of missing data in movement variables will be explored more thoroughly and other clustering algorithms will be tested.

To further explore the importance of cycles of iteration, the number of times participants ran the code on their robot might be detected from screencast recordings of the participant laptop. We do not have log files from Tinker to analyze, but computer vision algorithms should be able to detect how often the "run" button was pressed during a session. With the Kinect sensor no longer being produced, future work may rely solely on video recording with joints and coordinates determined by computer vision software rather than sensors. This would aid the scalability of these techniques by reducing the cost of implementation in classrooms and other learning environments.

8. REFERENCES

[1] Starr, E., Reilly, J., and Schneider, B. 2018, June. Using Multi-Modal Learning Analytics to Support and Measure Collaboration in Co-Located Dyads. Full paper to be presented at the International Conference on the Learning Sciences. London, England.

[2] Schneider, B., and Blikstein, P. 2015. Unraveling students' interaction around a tangible interface using multimodal learning analytics. *Journal of Educational Data Mining* 7, 3, 89-116.

[3] Dede, C. 2010. Comparing frameworks for 21st century skills. *21st century skills: Rethinking how students learn*, 20, 51-76.

[4] Grover, S. and Pea, R. 2013. Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42, 1, 38-43.

[5] Blikstein, P. and Worsley, M. 2016. Multimodal Learning Analytics and Education Data Mining: using computational technologies to measure complex learning tasks. *Journal of Learning Analytics*, 3, 2, 220-238.

[6] Berland, M., Baker, R.S. and Blikstein, P. 2014. Educational data mining and learning analytics: Applications to constructionist research. *Technology, Knowledge and Learning*, 19, 1-2, 205-220.

[7] Stahl, G., Koschmann, T., and Suthers, D. 2006. Computer-supported collaborative learning: An historical perspective. In R. K. Sawyer (Ed.), Cambridge handbook of the learning sciences, 409-426. Cambridge, UK: Cambridge University Press.

[8] Roschelle, J. and Teasley, S.D. 1995. The construction of shared knowledge in collaborative problem solving. In *Computer supported collaborative learning*, 69-97. Springer, Berlin, Heidelberg.

[9] Schneider, B. and Pea, R. 2013. Real-time mutual gaze perception enhances collaborative learning and collaboration quality. *International Journal of Computer-supported collaborative learning*, 8, 4, 375-397.

[10] Raca, M., Tormey, R. and Dillenbourg, P. 2014, March. Sleepers' lag-study on motion and attention. In *Proceedings of the Fourth International Conference on Learning Analytics and Knowledge*, 36-43. ACM.

[11] Schlömer, T., Poppinga, B., Henze, N. and Boll, S. 2008, February. Gesture recognition with a Wii controller. In *Proceedings of the 2nd International Conference on Tangible and Embedded Interaction*, 11-14.

[12] Grafsgaard, J., Wiggins, J., Boyer, K.E., Wiebe, E. and Lester, J., 2014, July. Predicting learning and affect from multimodal data streams in task-oriented tutorial dialogue. In *Proceedings of the 7th International Conference on Educational Data Mining*, 122-129.

[13] Worsley, M. and Blikstein, P. 2013. Towards the Development of Multimodal Action Based Assessment. *Proceedings of the Third International Conference on Learning Analytics and Knowledge,* 94–101. ACM.

[14] Howison, M., Trninic, D., Reinholz, D. and Abrahamson, D. 2011, May. The Mathematical Imagery Trainer: from embodied interaction to conceptual learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1989-1998.

[15] Roth, W.M. 2001. Gestures: Their role in teaching and learning. *Review of Educational Research*, 71, 3, 365-392.

[16] Meier, A., Spada, H., and Rummel, N. 2007. A rating scheme for assessing the quality of computer-supported collaboration processes. Computer Supported Learning. 2, 63–86.

[17] Brennan, K. and Resnick, M. 2012. New frameworks for studying and assessing the development of computational thinking. Presented at the Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada.

[18] Weintrop, D. and Wilensky, U. 2015. Using commutative assessments to compare conceptual understanding in blocks-based and text-based programs. Presented at the 11th Annual ACM Conference on International Computing Education Research. ICER.

[19] Chi, J.T., Chi, E.C. and Baraniuk, R.G. 2016. k-pod: A method for k-means clustering of missing data. *The American Statistician*, *70*, 1, 91-99.

[20] Tschan, F. 2002. Ideal cycles of communication (or cognitions) in triads, dyads, and individuals. *Small Group Research*, 33, 6, 615-643.

[21] Ponce-López, V., Escalera, S., & Baró, X. 2013, December. Multi-modal social signal analysis for predicting agreement in conversation settings. In Proceedings of the 15th ACM on International conference on multimodal interaction, 495-502. ACM.

[22] Won, A. S., Bailenson, J. N., Stathatos, S. C., and Dai, W. 2014. Automatically detected nonverbal behavior predicts creativity in collaborating dyads. Journal of Nonverbal Behavior, 38, 3, 389-408.

[23] Martinez-Maldonado, R., Dimitriadis, Y., Martinez-Monés, A., Kay, J., and Yacef, K. 2013. Capturing and analyzing verbal and physical collaborative learning interactions at an enriched interactive tabletop. *International Journal of Computer-Supported Collaborative Learning*, *8*, 4, 455-485.

[24] Schneider, B., Sharma, K., Cuendet, S., Zufferey, G., Dillenbourg, P., and Pea, R. 2016. Detecting collaborative dynamics using mobile eye-trackers. Presented at the 12th International Conference of the Learning Sciences, Singapore: International Society of the Learning Sciences.

[25] Graesser, A. C., McNamara, D. S., Louwerse, M. M., and Cai, Z. 2004. Coh-Metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, & Computers*, 36, 2, 193–202.

[26] Pijeira-Díaz, H. J., Drachsler, H., Järvelä, S., & Kirschner, P. A. (2016). Investigating collaborative learning success with physiological coupling indices based on electrodermal activity. Presented at the Proceedings of the sixth international conference on learning analytics & knowledge, 64–73. ACM.

Automated Speech Act Categorization of Chat Utterances in Virtual Internships

Dipesh Gautam
The University of Memphis
Memphis, TN 38152
dgautam@memphis.edu

Nabin Maharjan
The University of Memphis
Memphis, TN 38152
nmharjan@memphis.edu

Arthur C. Graesser
The University of Memphis
Memphis, TN 38152
graesser@memphis.edu

Vasile Rus
The University of Memphis
Memphis, TN 38152
vrus@memphis.edu

ABSTRACT

This work is a step towards full automation of auto-mentoring processes in multi-player online environments such as virtual internships. We focus on automatically identifying speaker's intentions, i.e. the speech acts of chat utterances, in such virtual internships. Particularly, we explore several machine learning methods to categorize speech acts, with promising results. A novel approach based on pre-training a neural network on a large set of (and noisy) labeled data and then on expert-labeled data led to best results. The proposed methods can help understand patterns of conversations among players in virtual internships which in turn could inform refinements of the design of such learning environments and ultimately the development of virtual mentors that would be able to monitor and scaffold students' learning, i.e., the acquisition of specific professional skills in this case.

Keywords

speech act, virtual internships, online tutoring, classification, neural networks, machine learning

1. INTRODUCTION

Virtual internships are simulations where interns gain professional experience while participating in an online fictional company. That is, they go through an internship experience without actually being present in a physical, actual company. In such virtual internships, the student interns participate in activities such as solving designated problems or tasks for which they actively interact with their mentor(s) as well as other interns through instant text messages, voice messages, chatrooms, and multimedia elements. The learning that occurs in engineering virtual internships, our focus, can be characterized by epistemic frame theory. This theory claims that professionals develop epistemic frames, or the network of skills, knowledge, identity, values, and epistemology that are unique to that profession [17]. For example, engineers share ways of understanding and doing (knowledge and skills); beliefs about which problems are worth investigating (values), characteristics that define them as members of the profession (identity), and a ways of justifying decisions (epistemology).

It is important to understand patterns of conversations between the various players in a virtual internship in order to refine the design of such virtual internships and to ultimately develop a virtual mentor that would be able to monitor and scaffold students' learning, i.e., the acquisition of specific professional skills in this case. Currently, virtual internship environments rely on human mentors. Our work here is a step towards a deeper understanding and full automation of the mentoring process. Indeed, understanding the mentoring process implies detecting patterns of actions by the mentor and by the students that are effective. Since conversations are the main type of interactions between the mentors and the student interns, understanding the actions or intents behind each utterance in the conversations is critical. We offer here such solutions to automatically detecting the intent, or speech act, behind chat utterances in virtual internships. Furthermore, such solution are critical to fully automate the mentoring process, i.e., to building automentors. Indeed, knowing students' speech acts can inform an automated mentoring agent to plan the best reply. For instance, if a student is greeting, the system should respond with a greeting or if a student is asking a question the system should plan to, for instance, answer the question.

Speech acts are 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. Speech act theory was developed based on the "language as action" assumption. The basic idea is that behind every utterance there is an underlying speaker intent, called the speech act. For instance, the utterance "Hello, John!" corresponds to a greeting, that is, the speaker's intention is to greet, whereas the utterance "Which web browser are you using?" is about asking a question. As already hinted earlier, discovering learners' patterns of actions in the form of patterns of (speech) acts in virtual internships could be revealing. For instance, we may find that interns that ask more questions acquire better and faster target professional skills based on the theory that asking more relevant questions indicates a more active and engaged learner which typically leads to more effective and efficient learning processes.

Labeling utterances with speech acts requires both an analysis of the utterance itself, e.g., "Hello" clearly indicates a greeting, but also accounting for the previous context, i.e., previous utterances in the conversation. For instance, after a question, a response most likely follows. This pattern holds in dialogues, i.e., interactions between two conversational partners where there is a clear pattern of turn-taking; that is, a speaker's turn is followed by a turn by the other speaker. However, in multi-player conversations such as the one that we deal with in this work, identifying the previous utterance that is most relevant to the current one is more difficult. For example, in the snapshot of conversation shown in Table 1 from one of our virtual internships, the question in chat utterance 3 from *player2* is addressed to the *mentor* whose reply is in utterance 6. The next *Player2's* reply is in utterance 9. Indeed, in such multi-party conversations, it becomes more challenging to link a target utterance to the previous one that triggered it. The complexity of untangling such multi-player conversations is further increased as the number of participants increases. Therefore, even though the speech act of an utterance is determined to some degree by the previous, related chat utterances, in this work we explore a method for speech act classification that relies only on the content of the target utterance itself, ignoring the previous context.

Table 1: A Snapshot of Conversation in Nephrotex

S.N.	Speaker	Utterance
1	mentor	I'm here to help you.
2	player1	hi!
3	player2	*Has anyone been able to get the tutorial notebook to open?*
4	player3	Hey
5	player4	Hello!
6	mentor	*Which web browser are you using?*
7	player3	are you guys real?
8	player1	yes we're real lol
9	player2	*I switched to Firefox, now everything is working. Thanks!*

To this end, we used various existing classifiers such as Naive Bayes and decision trees along with a Neural Network (NN) approach. Based on previous experience such as [15, 14], we selected leading words in each utterance as the features of the underlying model. The feature-based representations of utterances were then fed into Naive Bayes and decision tree classifiers. For neural networks, we used the pre-trained sent2vec[11] model, trained on a large collection of Wikipedia articles, to map an entire utterance onto a vector representation or embedding. Nevertheless, our data is dialogue data which differs from Wikipedia texts to some degree. To compensate for this discrepancy, the basic model is used to further train a small neural network using a comparatively small domain specific dataset in order to improve the predictive power for the type of instances seen in our dataset. That is, this is a form of transfer learning where our model first uses generic knowledge from the pre-trained Wikipedia model which is then tranferred or adapted to a specific domain data by training with domain data. Furthermore, using pre-trained models can also lead to better parameter learning in NN [12].

We also investigated a novel approach to building a speech act classifier for multi-player conversational systems using a mix of noisy and golden data, as explained next. In this approach, we trained a decision tree model with a small set of human annotated data and then used that trained model to generate (noisy) labels for a much larger collection of utterances. The noisy labeled utterances were then used to pre-train the neural network and then further trained with the human annotated gold dataset. The advantage of pre-training here is to have a huge collection of training data to pre-train the network and then refine the training using the (smaller) human-annotated (noise-free or gold) dataset.

Next, we present a quick overview of related work in this area before presenting details of our methods and experiments and resuts.

2. BACKGROUND

As mentioned, our approach to label utterances with speech acts is based on the speech act theory according to which when we say something we do something [1, 16]. Austin theorized the acts performed by natural language utterances. Later on, Searle[16] refined Austin's idea of speech acts by emphasizing the psychological interpretation based on beliefs or intentions. According to Searle, there are three levels of actions carried by language in parallel. First, there is the locutionary act which consists of the actual utterance and its exterior meaning. Second, there is the illocutionary act, which is the real intended meaning of the utterance, its semantic force. Third, there is the perlocutionary act which is the practical effect of the utterance, such as persuading and encouraging. In a few words, the locutionary act is the act of saying something, the illocutionary act is an act performed in saying something, and the perlocutionary act is an actperformed by saying something. For example, the phrase "Don't go into the water" might be interpreted at the three act levels in the following way: the locutionary level is the utterance itself, the morphologically and syntactically correct usage of a sequence of words; the illocutionary level is the act of warning about the possible dangers of going into the water; finally, the perlocutionary level is the actual persuasion, if any, performed on the hearers of the message, to not go into the water.

Many researchers have explored the task of automatically classifying speech acts as well as the related task of discovering speech acts. For instance, Rus and colleagues [14] proposed a method to automatically discover speech act categories in dialogues by clustering utterances spoken by participants in educational games. In our case, we use a predefined taxonomy of speech acts which was inspired by Rus and colleagues' work and further refined by dialogue experts.

The same group of researchers explored the role of Hidden Markov Models (HMMs), a generative model, and Conditional Random Fields (CRFs), a discriminative model, in classifying speech acts in one to one human tutorial sessions [13]. They demonstrated that the CRF model with features constructed from the first three tokens and last token of previous, next and current utterances, length of current utterance, and other surface features such as bigrams and the speech acts of context utterances performed better than HMM models. They have not worked with multi-party conversations as it is the case in our work.

In other work, Moldovan and colleagues [9] applied supervised machine learning methods to automatically classify chats in an online chat corpus. The corpus consisted of online chat sessions in English between speakers of different ages. Their supervised approach relied on an expert defined set of speech act categories. In their work, they hypothesized that the first few tokens were good predictors of chat's speech act. Samei et al. [15] adopted Moldovan's hypothesis about the predictive power of first few tokens and extended the supervised machine learning model with contextual information, i.e., previous and following utterances. From their experiments with data from an online collaborative learning game, they found that the role of context is minor and therefore context is not that important and can mostly be ignored in predicting speech acts. Similar to those works, we also explore the effectiveness of leading word tokens in utterances for Naive Bayes and decision tree based classifiers.

Ezen and Boyer [4] proposed an unsupervised method for dialogue act classification. They used a corpus from a collaborative learning programming tutor project which consisted of dialogues between pairs of tutors and students collaborating on the task of solving a programming problem. They applied an information retrieval approach in which the target utterance was considered as a query and the rest of the utterances were considered as documents. Based on the ranked list of relevant utterances to the query utterance, a vector representation is derived for each query utterance. The vector representation is then fed into a k-means clustering algorithm to identify clusters of utterances. For evaluation purposes, they used manually labeled data. Each cluster was assigned the majority human-generated label of all utterances in the cluster. An utterance that was placed in a particular cluster by the k-means clustering algorithm was assigned the label of that cluster as its speech act category for evaluation purposes. It should be noted that they varied the number of clusters to obtain a maximum overall accuracy of the discovered labels. Their algorithm outperformed a previous approach for dialogue act clustering, which Ezen and Boyer used for classification and which relied on a simple tf-idf representation and cosine similarity for clustering.

Kim and colleagues investigated the task of classifying dialogue acts in multi-party chats[8]. They analyzed two different types of live chats: (i) live forum chats with multiple participants from the US Library of Congress and (ii) Naval Postgraduate School (NPS) casual chats [5]. In order to classify the utterances in the chats in various speech act categories, Kim and colleagues [7] used speech act patterns which they defined manually using cue words derived from the utterances. They classified the discussion contributions into six speech act categories. They found that the previous chat utterances used as context did not contribute significantly to predicting speech acts in multi-party conversations until the entanglement amongst the utterances was resolved. Our work is similar to theirs in the sense that we analyze multi-party conversations. Nevertheless, our work is conducted in the context of the virtual internship Nephrotex, where learners focus on specific design problems as opposed to the types of conversations used by Kim and colleagues such as the casual NPS chats, which did not focus on a particular given task. We do not explore the accuracy of our methods in context. Furthermore, we do not resolve the entangled dialogues and then use contextual information for speech act classification. We do plan to address the role of context and entanglement in multi-party conversations in future work.

A regular expression based speech act classifier was proposed by Olney et al[10]. Their classifier used regular expression which they called a finite state transducer to classify utterances of AutoTutor, an intelligent tutoring system. They showed that the classifier constructed by cascading parts of speech information, the finite state transducer, and word sense disambiguation rules yielded good performance in classifying utterances into 18 categories. We have not compared our work with a regular expression based classifier due to the labor intensive aspects of such an approach. Typically, such regular-expression approaches should lead to high-precision results and not generalize very well unless they target speech act categories which are more or less closed-class such as greeting expressions (there is a limited number of expressions in which someone can greet).

3. ENGINEERING VIRTUAL INTERNSHIPS

Our work presented here was conducted on conversations among students and mentors in *Nephrotex* (NTX), a virtual internship. Nephrotex was designed and created to improve engineering undergraduate students' professional skills. It was incorporated into first-year engineering undergraduate courses at the University of Wisconsin-Madison[3].

In NTX, groups of students work together on a design problem, e.g. designing filtration membranes for hemodialysis machines, with the help of a mentor. Working on a design problem involves choosing design specifications from a set of input categories. Each student is assigned to a team of five members. There were five such teams who were each expected to learn about one of five different materials.

After completing a set of preliminary tasks, students design five prototypes to submit for testing. Later, they receive performance results for these prototypes which they have to analyze and interpret. Overall, students in each internship complete two such cycles of designing, testing, and analysis before deciding on a final design to recommend. During these cycles, students hold team meetings via the virtual internship's chat interface in which they reflect on their design process and make decisions on how to move forward. Once teams recommend a final design, they present this design to their peers. The conversations among the participants take place virtually via an online chat interface in Nephrotex, or in person outside of the class.

As previously mentioned, in this work, we focus on analyzing chat utterances in Nephrotex in order to discover the underlying speech act. Automated speech acts classification could have significant impact on scaling virtual internships to all students, anytime, anywhere via Internet-connected devices. This is not currently possible because the human mentors can only handle that much.

4. METHODS

Our approach to classifying learner utterances in virtual internships relies on machine learning algorithms that take as input utterances represented in a feature space. The features in our case are either surface features (such as leading words) or latent features (such as dimensions in neural sentence embeddings). We developed and compared the performance of two different categories of classifiers that rely on these two types of representations. We describe briefly those classifiers, the features we used, and the results obtained during experiments meant to validate the proposed classifiers.

4.1 Classifier Using Surface Features

The surface feature representation of a text uses a number of important lexical and syntactic elements such as leading words or the punctuation mark at the end of the utterance, e.g., the ending question mark at the end of a question. In conversation data such as chat utterances in virtual internship, lexical features such as leading words alone have competitive power in terms of speech act representation of the utterance. Therefore we adopted the model representation proposed previously [9, 14] due to its solid theoretical foundations and competitive results. The basis of this approach is that humans infer speakers' intention after hearing only few of the leading words of an utterance. One argument in favor of this assumption is the evidence that hearers start responding immediately (within milliseconds) or sometimes before speakers finish their utterances[6]. Accordingly, we selected few leading words (first few words) of the utterance as the features to represent the utterance. Although we have experimented with different number of leading words, we report here results with the six leading words (first six words) as this combination yielded best performance as explained later. Once each utterance was mapped onto such a feature-representation, we performed experiments with two different types of classifiers: naive Bayes and decision trees.

Before feature construction, we pre-processed the utterances by lemmatizing the words and removed the punctuations. Although some of the punctuations, such as "*question mark (?)*" or "*exclamation mark (!)*", are predictive on some of the speech acts, they seem to not always be present in or seem to appear at improper places in the utterance. Hence we ignored the punctuations for our analysis here.

4.2 Classifier Using Latent Features

The other category of classifiers we used relies on latent features that were automatically learned using neural networks. These features are the components of automatically generated vectors that represent sentences. Such neural network generated vectors are derived from textual units such as character, letter n-grams, words and words n-grams. In our model, we adopted sent2vec, a sentence representation model proposed by Pagliardini and colleagues [2, 11] and which was developed by training a neural network on a collection of Wikipedia articles.

Based on such latent representations of utterances, we designed a neural network model in two stages. First, the model obtained a latent representation for an utterance using the generic pre-trained sent2vec model. In a second stage, the embedded vector representation is used to further train our neural network to perform speech act classification.

Table 2: Speech Act Taxonomy with Examples

Speech Acts	Examples
expressive evaluation (xpe)	–It is excellent in all values except for cost –great –The lag is pretty bad
greeting (gre)	–Welcome back interns ! –Hello Team !
metastatements (mst)	–sorry littles confused here –Whoops , I was reading that wrong . –lol
other (oth)	–or addition –etc
question (que)	–Is biocompatability cummulative ? –who is going to write the email ?
reaction (rea)	–I 'm ok with this –alright , i think i agree with u guys
request (req)	–Please keep that in mind during your team selection of membrane prototypes . –K , I would like to start the team meeting now .
statement (stm)	–I read an article that said most dialyzers take 6 hours to run . –I can start the meeting with jamon ...

While training the neural network with domain specific data, we applied two methods of training. In the first method, we used a small set of human annotated gold data for training and validation. In the second method, we pre-trained the neural network with noisy labeled data generated from a domain corpus and then further trained and validated the model with gold data. We will discuss in detail the process of generating noisy labels in the next section.

5. EXPERIMENTS AND RESULTS

In this section, we present the experiments that were conducted and the results obtained, starting with a brief description of the data we used.

Table 3: Distribution of Speech Acts in Corpus

	Human Labeled		Noisy Labeled	
Speech act	#	%Dist	#	%Dist
expressive evaluation	24	2.4	256	1.26
greeting	14	1.4	285	1.40
metastatements	40	4.0	405	2.00
other	11	1.1	166	0.82
question	173	17.3	3098	15.25
reaction	202	20.2	3347	16.47
request	56	5.6	1041	5.12
statement	480	48.0	11719	57.68
Total	1000		20317	

5.1 The Virtual Internship Conversation Dataset

Our dataset consists of a collection of more than 22 thousands utterances from the Nephrotex virtual internship. The eight categories of speech acts we used are presented in Table 2 (acronyms are shown in parentheses) together with example utterances.

From the examples, it could be observed that the leading tokens in each utterance are indicative of the underlying speech act shown in the first column. For instance, *greetings* start with "Hello" and "Welcome back" whereas *questions* start with wh-words ("Who") or auxiliary verbs ("Is") while requests start with "Please", which is typically used to ask for something in a nice manner.

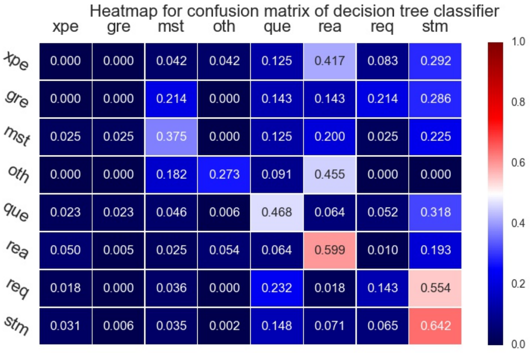


Figure 1: Confusion matrix for classification of decision tree (values refer to percentage expressed in decimal, acronyms refer to the speech acts defined in table 2)

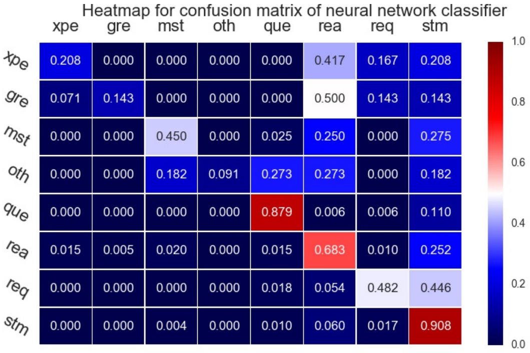


Figure 2: Confusion matrix for classification of neural network (values refer to percentage expressed in decimal, acronyms refer to the speech acts defined in table 2)

5.1.1 The Data Annotation Process

Of the 22,317 utterances, 2,000 utterances were manually annotated by three annotators. Out of these 2,000 utterances, 1,000 utterances were used for training the annotators. Agreement among annotators was computed as the average of Cohen's kappa between all possible pairs of annotators. The average agreement between any two annotators was 0.64.

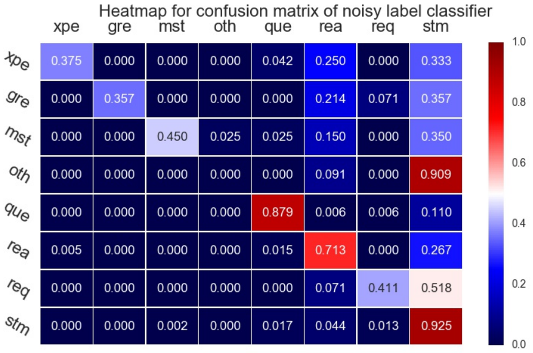


Figure 3: Confusion matrix for classification of noise label trained neural network (values refer to percentage expressed in decimal, acronyms refer to the speech acts defined in table 2)

The remaining 1,000 utterances were labeled by the annotators after finishing their training. The average agreement, measured as Cohen's kappa, among the coders was 0.69. To generate a final, unique label for each annotated utterance in cases in which there were any disagreements, a discussion among the annotators took place as well as the group of co-workers in the project team. We used the 1,000 human-labeled utterances as a gold dataset on which a 10-fold cross validation evaluation methodology was applied to evaluate the proposed speech act classification methods.

5.1.2 The Noisy Label Generation

The rest of the utterances in the whole dataset of 22,317 utterances was automatically labeled using the decision tree model trained on the first 1,000 instances labeled by trainee annotators. We chose decision trees to generate noisy labels because decision trees performed better than the Naive Bayes classifier. It should be noted that we used the other 1000 human-labeled gold data for 10 folds cross validations of our classifier models. Table 3 shows the distribution of speech acts in the gold and noisy labeled datasets. From the table, we observe that the noisy labels generated follow roughly comparable pattern of distribution for the speech acts that are more frequent in corpus. Therefore it makes sense to some extent to use those noisy labels to pre-train the neural network model.

5.2 Results

The results of the 10-fold cross-validation evaluation are summarized in Table 4 and Table 5. We report performance in terms of precision, recall, F-1 score, accuracy, and kappa. The data in Table 4 suggests that the performance of the neural network classifier is highest of all with an average F-1 score and accuracy of 0.764 and 0.779, respectively, and kappa of 0.666, which are the highest among all three types of classifiers including Naive Bayes and decision trees. Moreover, the two sample t-test on 10-fold cross validation accuracies revealed that, neural network performed significantly better than Naive Bayes ($p - value \approx 0.00$) and decision tree with ($p - value \approx 0.00$).

The results shown in Table 5 shows that the neural network model pre-trained with noisy labels improved the per-

Table 4: Performance of Naive Bayes, Decision Tree and Neural Network Classifiers

	NB			DT			NN		
Speech Act	P	R	F1	P	R	F1	P	R	F1
expressive evaluation	0.200	0.042	0.069	0.000	0.000	0.000	0.556	0.208	0.303
greeting	1.000	0.143	0.250	0.000	0.000	0.000	0.667	0.143	0.235
metastatements	0.000	0.000	0.000	0.283	0.375	0.323	0.692	0.450	0.545
other	0.099	1.000	0.180	0.176	0.273	0.214	1.000	0.091	0.167
question	0.000	0.000	0.000	0.429	0.468	0.448	0.921	0.879	0.899
reaction	0.354	0.342	0.348	0.630	0.599	0.614	0.687	0.683	0.685
request	0.000	0.000	0.000	0.143	0.143	0.143	0.614	0.482	0.540
statement	0.581	0.831	0.684	0.680	0.642	0.660	0.791	0.908	0.846
Weighted Average	0.370	0.482	0.406	0.549	0.536	0.542	**0.774**	**0.779**	**0.764**
	Accuracy = 0.482			Accuracy = 0.536			Accuracy = **0.779**		
	Kappa = 0.177			Kappa = 0.341			Kappa = 0.666		

Table 5: Performance of Noise Label Trained Neural Network Classifier

Speech Act	P	R	F1
expressive evaluation	0.900	0.375	0.529
greeting	1.000	0.357	0.526
metastatements	0.947	0.450	0.610
other	0.000	0.000	0.000
question	0.921	0.879	0.899
reaction	0.774	0.713	0.742
request	0.742	0.411	0.529
statement	0.762	0.925	0.835
Weighted Average	**0.796**	**0.795**	**0.781**
Accuracy	**0.795**		
Kappa	0.685		

formance. The overall improvement in precision, recall, F-1 score, and accuracy is about 2% with about 2% better kappa when compared to the neural network classifier (Table 4) without using the much larger, noisy label dataset. However, a t-test showed that the accuracy of the noisy label trained neural network is not significantly better than neural network trained without noisy label data $(p-value \approx 0.53)$. This could have happened because of the small samples used for the t-test: 10 from 10-folds cross validations. Using a larger number of folds, say, 50, could help us getting a large sample of accuracy values. It can be observed from the table that the performance for the "*other*" category is the weakest among all four classifiers. The reason is because of the nature of those utterances which contain only a few tokens, i.e., one or two words (see Table 2), with a lot of variation in terms of lexical content. In addition, the human labeled dataset contained few instances for this category which resulted in poor performance when the neural network model was trained using the human labeled data. Similarly, in the noisy, automatically-labeled dataset there are many misclassified "*other*" instances which led to poor training of the neural network model. Furthermore, the next phase of training the pre-trained neural network model with the human labeled data did not compensate enough because there were not sufficient "other" instances in the human labeled data to correct the pre-trained model. This is further supported by analyzing the confusion matrix where the number of true positives for the "*other*" category is 0%; the "*other*" category is labeled as "*statement*" 90% of the time in the case of the neural network model pre-trained with noisy labels (see Figure 3). Further evidence for this is provided by analyzing the confusion matrix for neural network trained only with gold labels where true positives for "*other*" utterances was 9% (see Figure 2). In this case, "other" utterances were labeled as "*question*" and "*reaction*". Other challenging speech acts are "*request*", which is most often confused with "*statement*". This is not surprising as the lexical composition of requests and statements is similar to some degree.

For decision trees, a quick analysis of the confusion matrix (see Figure 1) revealed that the true positives for "*expressive evaluation*" was 0%, being confused mostly with "*reaction*" or "*statement*" (41% and 29% of the time, respectively). Also, "*greeting*" is confused with "*metastatement*" by 21%, "*request*" by 21%, and "*statement* by 28%".

6. CONCLUSIONS

In this work, we explored several methods for speech act classification. We explored various classifier models with different categories of features as well as training strategies. We found that the latent features generated by a pre-trained sentence embeddings model (derived from a large Wikipedia corpus) yielded better performance compared to the other models. Besides that, the predictive power of the neural network model was further boosted when pre-trained with noisy label before training with expert-annotated data.

In future work, we plan to expand the current models by using more contextual information. Given the multi-party nature of our conversation data, before we can use contextual information, it is necessary to disentangle the conversations into sets of related utterances. Our future models will disentagle the multi-party conversations before attempting to use contextual information for speech act classification.

7. ACKNOWLEDGMENTS

This work was funded in part by the National Science Foundation (DRL-0918409, DRL-0946372, DRL-1247262, DRL-1418288, DRL-1661036, DRL-1713110, DUE-0919347, DUE-1225885, EEC-1232656, EEC-1340402, REC-0347000), the MacArthur Foundation, the Spencer Foundation, the Wisconsin Alumni Research Foundation, and the Office of the Vice Chancellor for Research and Graduate Education at the University of Wisconsin-Madison. The opinions, findings, and conclusions do not reflect the views of the funding agencies, cooperating institutions, or other individuals.

8. REFERENCES

[1] Austin, J.L. 1962. *How to do Things with Words.* Oxford University Press.

[2] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. 2016. Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606. Retrived from https://arxiv.org/abs/1607.04606

[3] D'Angelo, C. M., Arastoopour, G., Chesler, N. C., & Shaffer, D. W. 2011. Collaborating in a virtual engineering internship. In *Computer Supported Collaborative Learning Conference.* Hong Kong SAR, Hong Kong, China, 4-8.

[4] Ezen-Can, A., & Boyer, K. E. 2013. Unsupervised classification of student dialogue acts with query-likelihood clustering. In *Educational Data Mining.*

[5] Forsyth, E. N. 2007. *Improving automated lexical and discourse analysis of online chat dialog.* Doctoral dissertation. Naval Postgraduate School, Monterey, California.

[6] Jurafsky, D., & Martin, J. H. 2009. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.* Prentice Hall series in artificial intelligence, p.814.

[7] Kim, J., Chern, G., Feng, D., Shaw, E., & Hovy, E. 2006. Mining and assessing discussions on the web through speech act analysis. In *Proceedings of the Workshop on Web Content Mining with Human Language Technologies at the 5th International Semantic Web Conference.*

[8] Kim, S. N., Cavedon, L., & Baldwin, T. 2012. Classifying dialogue acts in multi-party live chats. In *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation.* 463-472.

[9] Moldovan, C., Rus, V., & Graesser, A. C. 2011. Automated Speech Act Classification For Online Chat. *MAICS.* 710, 23-29.

[10] Olney, A., Louwerse, M., Matthews, E., Marineau, J., Hite-Mitchell, H., & Graesser, A. 2003. Utterance classification in AutoTutor. In *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing.* 2. Association for Computational Linguistics, 1-8.

[11] Pagliardini, M., Gupta, P., & Jaggi, M. 2017. Unsupervised learning of sentence embeddings using compositional n-gram features. arXiv preprint arXiv:1703.02507. Retrived from https://arxiv.org/abs/1703.02507

[12] Pan, S. J., & Yang, Q. A survey on transfer learning. 2010. *IEEE Transactions on knowledge and data engineering.* 22(10), 1345-1359.

[13] Rus, V., Maharjan, N., Tamang, L. J., Yudelson, M., Berman, S., Fancsali, S. E. & Ritter, S. 2017. An Analysis of Human Tutors' Actions in Tutorial Dialogues. In *Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference.* AAAI, 122-127.

[14] Rus, V., Moldovan, C., Niraula, N., & Graesser, A. C. 2012. Automated Discovery of Speech Act Categories in Educational Games. *International Educational Data Mining Society.*

[15] Samei, B., Li, H., Keshtkar, F., Rus, V., & Graesser, A. C. 2012. Context-based speech act classification in intelligent tutoring systems. In *International Conference on Intelligent Tutoring Systems.* Springer, Cham, 236-241.

[16] Searle, J.R. 1969. *Speech Acts.* Cambridge University Press, GB.

[17] Shaffer, D. W. 2006. *How Computer Games Help Children Learn.* Macmillan.

Clustering the Learning Patterns of Adults with Low Literacy Skills Interacting with an Intelligent Tutoring System

Ying Fang[1], Keith Shubeck[1], Anne Lippert[1], Qinyu Cheng[1], Genghu Shi[1], Shi Feng[1], Jessica Gatewood[1], Su Chen[1], Zhiqiang Cai[1], Philip Pavlik[1], Jan Frijters[2], Daphne Greenberg[3] and Arthur Graesser[1]

[1]University of Memphis, [2]Brock University, [3]Georgia State University

{yfang2, kshubeck, alippert, qcheng, gshi, sfeng, jdgatewood16, schen4, zcai, ppavlik, grasser}@memphis.edu, jfrijters@brocu.ca, dgreenberg@gsu.edu

ABSTRACT

A common goal of Intelligent Tutoring Systems (ITS) is to provide learning environments that adapt to the varying abilities and characteristics of users. To do this, researchers must identify the learning patterns exhibited by those interacting with the system. In the present work, we use clustering analysis to capture learning patterns in over 250 adults who used the ITS, *CSAL* (Center for the Study of Adult Literacy) *AutoTutor*, to gain reading comprehension skills. AutoTutor has conversational agentsth at teach literacy adults with low literacy skills comprehension strategies in 35 lessons. These comprehension strategies align with one or more of the following levels specified in the Graesser-McNamara theoretical framework of comprehension: *word, textbase, situation model* and *rhetorical structure*. We used the adult learners' average response times per question and performance across lessons to cluster the students' learning behavior. Performance was measured as the proportion of 3-alternative-response questions answered correctly. Lessons were coded on one of the four theoretical levels of comprehension. Results of the cluster analyses converged on four types of learners: proficient readers, struggling readers, conscientious readers and disengaged readers. Proficient readers were fast and accurate; struggling readers worked slowly but were not accurate; conscientious readers worked slowly and performed comparatively well; disengaged readers were fast but did not perform well. Interestingly, the behaviors of learners in different clusters varied across the four theoretical levels. Identifying types of readers can enhance the adaptivity of AutoTutor by allowing for more personalized feedback and interventions designed for particular learning behaviors.

Keywords

CSAL; AutoTutor; Adult reader; Learner clustering; Intelligent Tutoring; Personalized Instruction

1. INTRODUCTION

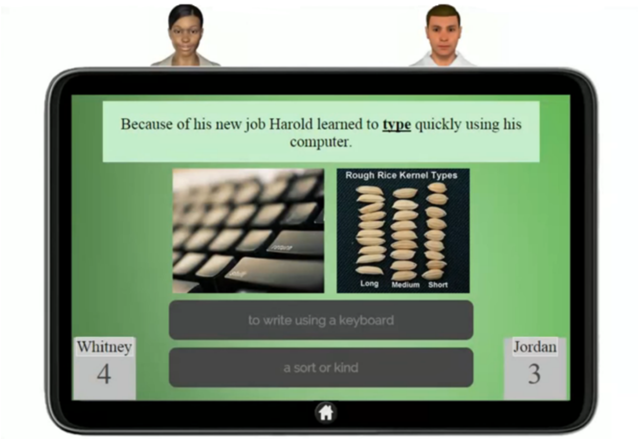
1.1 AutoTutor

AutoTutor is a conversation-based intelligent tutoring system (ITS) that has promoted learning on a wide range of topics [9, 13, 22]. AutoTutor, on average, has shown learning gains of 0.8 σ [22] compared to various traditional teaching controls. AutoTutor holds a conversation with students following an expectation-misconception tailored (EMT) approach [11]. This is a tutoring dialogue made up of questions that assess a learner's understanding of the content by comparing it to expected answers or misconceptions in real time. Using this EMT approach, AutoTutor is constantly assessing the students by providing feedback, hints, pumps, prompts to guide learning of the content.

Traditional AutoTutor systems implement conversations called *dialogues* that model the interactions that occur between a single human tutor and human student. More recent versions of AutoTutor oftenemploy *trialogues* which are tutorial conversations between three actors: a teacher agent, a human learner, and a peer agent [10, 12]. Trialogues offer several affordances over dialogues. For example, in a trialogue setting, the human learner can model productive learning behaviors that are programmed into the peer agent. The peer agent may also express misconceptions that the human learner shares and the negative feedback received from the tutor agent can be directed to the peer agent instead of the human learner. This helps avoid many of the undesirable effects from receiving direct negative feedback. Trialogues help students master difficult material. For example, trialogues successfully helped students learn scientific reasoning skills in an AutoTutor offshoot called Operation ARA [20, 21].

Agent trialogues are implemented in *AutoTutor* for CSAL [9], an ITS developed at the Center for the Study of Adult Literacy (CSAL, http://csal.gsu.edu/content/homepage). The web-based system is designed to help adults with low literacy acquire strategies for comprehending text at multiple levels of language and discourse. The system includes two computer agents (a teacher agent and a peer agent) which have conversations with human learners and between themselves. The learners are guided through their learning process by the computer agents. These three-way conversations are designed to (a) provide instruction on reading comprehension strategies, (b) help the learner apply these strategies to particular texts, (c) assess the learner's performance on applying these strategies, and (d) guide the learner in using the digital facilities. While previous implementations of AutoTutor relied on written natural language input from the learner, the learners in AutoTutor for CSAL have difficulties with writing. Thus, this version of AutoTutor was designed so that students interact through point-and-click, answering multiple choice questions, or using drag-and-drop. The conversational feature of

AutoTutor still guides the learner, but the questions can be solved without typed input. The lessons typically start with a 2-3 minutes video that reviews a comprehension strategy. After the review, the computer agents scaffold students through the learning by asking questions, providing short feedback, explaining how the answers are right or wrong, and filling in gaps of information. Figure 1 is an example of the teacher agent (on the left) asking both the learner and the peer agent (on the right) to find the meaning of the word "bank" in the given context. The scores of both the human learner and peer agent are shown under their names. The learner chooses the answer by clicking while the peer agent gives his answer by talking.



Because of his new job Harold learned to **type** quickly using his computer.

Figure 1: Example trialogue with competition which focuses on the meaning of words from context.

1.2 Theoretical Framework of Comprehension

The 35 lessons within AutoTutor align with the multilevel theoretical framework of comprehension proposed by Graesser and McNamara [13]. Six levels of comprehension were identified in Graesser and McNamara's framework. They are words, syntax, textbase, situation model, rhetorical structure/discourse genre, and pragmatic communication [13]. In this study, we focus on four of the six levels: *word, textbase, situation model and rhetorical structure.* The word level includes morphology, vocabulary and word decoding. The textbase level focuses on the explicit ideas in the text, but not the precise wording and syntax. The situation model refers to the subject matter content described in the text, including inferences activated by the explicit text. The model varies based on text type. In narrative text, the situation model includes the characters, objects, settings, events and other details of the story. In informational text, the model corresponds to substantive subject matter such as topics and domain knowledge. Rhetorical structure/discourse genre focuses on the category of text, such as narration, exposition, persuasion, and description. The word level represents the lower-level basic reading components, while the textbase, situation model and rhetorical structure level cover discourse components which were assumed to be more difficult to master [5, 24, 25].

1.3 Approaches of Categorizing Learners

A common goal of the learning sciences is to categorize learners based on their cognitive, motivational, and affective states. In the ITS domain, this is referred to as student modeling [23]. Student modeling is largely what enables ITS to be adaptive, with systems being designed to incorporate information pertaining to particular user characteristics. Specifically, ITS designers know that some specific cognitive states or behaviors are associated with learning and ensure the ITS can detect and respond appropriately to those features. Data mining approaches are often used to identify these attributes. For example, the ITS *Cognitive Tutor* employed a classifier to detect "gaming-the-system" behavior which occurred when users intentionally misused features of an ITS to progress through the content [1]. In another study that used data from students interacting with ALEKS (an ITS designed for math and science education), researchers were able to classify the learning persistence of a user as one of three distinct types [8]. Similarly, Del Valle and Duffy [7] clustered learners by their learning strategies in on online course and identified three types of learners: self-driven students, "get-it-done" students, and procrastinators. In another study, Wise et al. [27] clustered learners' online listening behaviors, and found three types of listeners with distinct behavioral patterns: superficial listeners, broad listeners and concentrated listeners.

In addition to categorizing or identifying learning behaviors from interacting with a system, researchers also categorize students based on individual differences in skill or knowledge gained a priori certain educational interventions. For example, in the ITS domain, students are often assessed on their prior knowledge of the domain material before interacting with an ITS, or at the early stages of the ITS content. They are commonly classified as having either high domain knowledge or low domain knowledge. There is evidence that high versus low domain knowledge students interact with ITS differently and require different pedagogical approaches to effectively learn from them. For example, an ITS using a vicarious learning design may benefit high domain knowledge students less than low-domain knowledge students [6]. This supports the idea that students with low-domain knowledge benefit more by observing peer agents or virtual tutees interacting with tutor agents. There is also evidence that high domain knowledge students sometimes suffer from an "expertise reversal effect" when presented with content they already understand [18]. When equipped with information about a learner's level of domain knowledge, ITS can leverage different pedagogical strategies to best cater to that student's capabilities.

The present study utilizes clustering analysis to achieve two goals. First, we characterize the behaviors of adults with low literacy skills who interacted with AutoTutor. Second, we examine whether adult readers' learning behaviors are associated with the different reading comprehension levels described above.

2. DATASETS AND DATA PROCESS

2.1 Data Sets

The data sets used for this study were taken from three waves of an intervention study consisting of 253 adult learners. The students participated in approximately 100 hours of hybrid classes which consisted of teacher-led sessions and AutoTutor sessions. The students took the Woodcock Johnson III Passage Comprehension subtest [28] before and after the intervention. While studying with AutoTutor, the logs of students' online learning activities were recorded by the system. The log file included learner information, class information, lesson and question information, response time and learning outcome.

In the intervention studies, 26 out of 35 AutoTutor lessons were assigned to the students; these 26 lessons were used for our analyses. We coded theoretical level of the lessons according to their primary theoretical levels. The classification of the primary theoretical levels is based on four discrete levels: Word, Textbase, Situation Model and Rhetorical Structure. The major components of Word lessons are word parts, word-meaning clues, learning new words and multiple meaning words. The Textbase lessons focus on pronouns, punctuation, key information, and main ideas. The Situation Model lessons mainly cover nonliteral language, connecting ideas, and inferences from text. The Rhetorical Structure lessons covered purpose of texts, steps in procedures, problems and solutions, cause and effect, compare and contrast, time and order, and other categories of rhetorical composition. In each lesson there are 10 to 35 questions. The questions in most lessons fall into two different difficulty levels. Normally a lesson starts with 10-15 medium level questions. Depending on the performance of the medium level questions, the learners are branched into hard or easy level questions in that lesson.

2.2 Data Process

First, we removed hard and easy questions so that only medium level questions were included in our statistical analyses. The reason for removing easy and hard questions was that response time was an important measure in the analysis, and response time could be confounded by using different question difficulty levels. Second, we removed motivational items; a motivational item was defined as any item that all the students answered correctly. These items could not be used for discriminating students and therefore they were removed from the analysis. Third, we examined the response time on each question and removed the outliers. According to the experimenters, the adult students infrequently took long breaks without logging out the system for various reasons, which led to some observations with extremely long response time. Following the rule of thumb about extreme outliers [19], we removed the response time which was three IQR (i.e. interquartile range) higher than the third quartile. For the lower end, the rule did not apply, so we replaced the bottom 1% of the observations with response times between 0 and 2 seconds with 3 seconds. The original log file had 102,519 observations. After data screening and cleaning, there were 42,289 observations from 253 students in dataset.

Next, we aggregated the data to student level and created variables for analyses. The aggregation was performed twice and two sets of features were created for analyses using the process described below.

In the log file, each observation represents an attempt that a student made on answering a question. All the students attempted multiple lessons, and within each lesson there were multiple questions, so each student had multiple observations in the log. The variables we used for the aggregations were the system-generated student ID, theoretical level of lessons, response time, and learning outcome. Each lesson was coded with a specific theoretical level (Word, Textbase, Situation Model or Rhetorical Structure) and the questions within the lesson were specific to the lesson's level. Response time was the time the learner spent working on the question, excluding the reading time. Learning outcome was either correct or incorrect. We aggregated the data based on these variables and calculated each student's average response time and accuracy at the four theoretical levels. After aggregation, the observations for each student were decreased to eight. The eight observations represented the average response time and accuracy at Word, Textbase, Situation Model and Rhetorical Structure levels. Response time was initially measured in seconds, which was a continuous variable. This measure remained the same after aggregation. Accuracy was a binary variable (i.e. 1 or 0) initially, but it became a continuous proportion correct variable after aggregation. Next, we changed the data format and combined the eight observations associated with one student into one observation with eight features. After this, there were 253 observations and each observation represented one student. The eight features were response time for Word, Textbase, Situation Model, and Rhetorical Structure level items, as well as the proportion correct for Word, Textbase, Situation Model, and Rhetorical Structure level items. This was how we created the first set of features. For the second feature set, we split response time into response time on correct answers and incorrect answers. Therefore, the response time features doubled from four to eight and the number of performance features remained four. Put together, we created two sets of features through aggregation. The first set had eight features and the second had twelve.

3. DATA EXPLORATION

Before data mining was carried out, we examined the student sample's response time and accuracy as a whole at the four theoretical levels to see whether response time was associated with theoretical level. The mean response time and accuracy at each level is shown in Table 1.

Table 1: Means and standard deviations of response time and accuracy at four theoretical levels.

	Response Time	Response Time (Correct)	Response Time (Incorrect)	Accuracy
Word	34.31 (σ = 23.55)	32.53 (σ = 12.91)	36.73 (σ =16.71)	0.67 (σ = 0.47)
Textbase	35.15 (σ = 23.38)	34.06 (σ = 11.23)	40.91 (σ = 17.44)	0.65 (σ = 0.48)
Situation Model	30.28 (σ = 22.81)	28.18 (σ = 9.15)	36.29 (σ = 13.58)	0.69 (σ = 0.46)
Rhetoric Structure	31.43 (σ = 23.95)	29.11 (σ = 11.10)	38.87 (σ = 12.66)	0.69 (σ = 0.46)

One-way ANOVAs were conducted to compare the means of response time, response time on correct items, response time on incorrect items and accuracy between the four theoretical levels. Results of the ANOVAs indicated that there were no significant differences between the four theoretical levels on response time or accuracy (F (3, 996) = 1.90, $p = 0.129$). However, we found theoretical level of the text affected both the time to give a correct response (F (3, 996) = 17.75, $p < 0.001$), and the time to give an incorrect response (F (3, 996) = 6.02, $p < 0.001$). Post hoc comparisons using the Tukey HSD test indicated that the average response time on correct attempts was longer at Word and Textbase levels than that of Situation Model and Rhetoric Structure levels. The average time on incorrect attempts at Textbase level was higher than that of Word and Situation Model levels. Since the differences found in response time on correct answers and incorrect were not consistent and did not show any pattern, we decided to group the students through clustering to investigate if theoretical levels influenced adult learners in a more nuanced way.

4. CLUSTER ANALYSES

Cluster analysis is a statistical exploratory tool used to find similar groups in an unsupervised fashion. It partitions objects into clusters so that the objects in the same cluster are more similar to each other than to those in other clusters. In educational settings, successful clustering has been achieved and the researchers identified learner groups with different behavioral patterns [3, 7, 27]. For example, Wise et al. [27] clustered learners' online listening behaviors and found three types of listeners with distinct behavioral patterns: superficial listeners, broad listeners and concentrated listeners. A similar goal can be transferred to our current context, with clustering possibly identifying groups with different learning behaviors across the four theoretical levels.

4.1 K-means Cluster Analysis

To carry out our clustering analysis, we applied a k-means clustering algorithm to our data. K-means clustering fits data points into clusters by iteratively reassigning and re-averaging the cluster centers until the points have reached convergence [15,16]. It is a common choice for clustering data since it is simple, effective and relatively efficient. We used R (version 3.3.3) to group students according to the k-means clustering algorithm of Hartigan and Wong [15].

Table 2: Cluster means and standard deviations on the eight features.

	Cluster 1 (n = 64)	**Cluster 2 (n = 45)**	**Cluster 3 (n = 88)**	**Cluster 4 (n = 53)**
Time (Word)	24.07 (σ = 7.02)	46.21 (σ = 13.27)	35.40 (σ =9.32)	31.33 (σ = 8.37)
Time (Textbase)	25.80 (σ = 5.33)	51.31 (σ = 9.98)	36.15 (σ =5.87)	34.41 (σ = 7.76)
Time (Situation)	22.40 (σ = 4.48)	43.87 (σ = 7.51)	29.76 (σ =4.90)	31.57 (σ = 7.76)
Time (Rhetorical)	22.10 (σ = 4.81)	44.36 (σ = 7.77)	31.86 (σ =5.06)	32.72 (σ = 7.36)
Accuracy (Word)	0.73 (σ = 0.13)	0.69 (σ = 0.14)	0.71 (σ = 0.13)	0.54 (σ = 0.17)
Accuracy (Textbase)	0.74 (σ = 0.13)	0.70 (σ = 0.18)	0.71 (σ = 0.12)	0.54 (σ = 0.12)
Accuracy (Situation)	0.73 (σ = 0.11)	0.65 (σ = 0.08)	0.74 (σ = 0.07)	0.59 (σ = 0.11)
Accuracy (Rhetorical)	0.75 (σ = 0.08)	0.72 (σ = 0.09)	0.71 (σ = 0.07)	0.61 (σ = 0.10)

Our choice to start with K=4 was guided by previous research. We assumed both engagement and disengagement existed while adult learners interacted with AutoTutor. For disengagement, a recent study on AutoTutor reported three types of behaviors associated with disengagement [14]. For engagement, another study used personalized time on item as a classifier, which was regard as a single type of behavior [21]. Put together, we assume there were four types of predominant behaviors that separate the learners into 4 clusters.We performed k-means clustering with k=4 twice: once with eight features and once with twelve features. As explained in section 2.2 (Data Process), the twelve features were developed from the eight features by dividing response time into response time on correct answers and incorrect answers. We also experimented with k = 3 and k = 5 and using the two feature sets. Compared with the 4-cluster solution, the 3-cluster solution lost some meaningful information. In the 5-cluster solution, two clusters had similar patterns. Therefore, we selected 4 as the optimum number of clusters. The results of the 4-cluster solution using eight features and twelve features are shown in Table 2 and Table 3, respectively.

We compared the 4-cluster solutions using 8 features with the one using 12 features. The four clusters showed similar patterns in the two solutions. The results of ANOVAs and post hoc tests comparing cluster differences on the grouping variables indicated similar between-cluster differences on response time and accuracy. We also tried k=3 and k=5 clustering, and compared the solutions from 8 features to that from 12 features. Both results indicated the consistency between solutions using 8 and 12 features. We further conducted Pearson correlation on the time variables (i.e. response time at different theoretical levels) with split time variables (i.e. response time on correct attempts and incorrect attempts at different theoretical levels). The results indicated significant moderate to strong correlations between these variables. The comparisons and statistical analyses suggested that splitting response time into two features did not contribute much to the discovery of the underlying structure. Following the principle of parsimony, we selected 8 features over 12 features for further analyses.

Table 3: Cluster means and standard deviations on the twelve features

	Cluster 1 (n = 63)	**Cluster 2 (n = 52)**	**Cluster 3 (n = 54)**	**Cluster 4 (n = 81)**
Time on Correct (Word)	24.08 (σ = 6.75)	44.57 (σ = 16.03)	35.47 (σ =8.11)	30.05 (σ = 9.15)
Time on Incorrect (Word)	23.78 (σ = 9.50)	47.59 (σ = 15.87)	48.03 (σ =14.67)	31.55 (σ = 8.91)
Time on Correct (Textbase)	24.98 (σ = 5.53)	46.47 (σ = 10.36)	32.95 (σ =6.41)	33.90 (σ = 8.60)
Time on Incorrect (Textbase)	31.23 (σ = 11.12)	57.05 (σ = 19.39)	43.25 (σ =15.59)	36.79 (σ = 8.13)
Time on Correct (Situation)	22.39 (σ = 3.84)	39.78 (σ = 8.59)	26.80 (σ =4.53)	27.71 (σ = 5.46)
Time on Incorrect (Situation)	27.35 (σ = 8.69)	50.84 (σ = 10.84)	34.91 (σ =11.17)	34.82 (σ = 10.15)
Time on Correct (Rhetorical)	20.04 (σ = 5.17)	41.47 (σ = 10.24)	27.55 (σ =4.76)	28.58 (σ = 5.69)
Time on Incorrect (Rhetorical)	29.11 (σ = 7.19)	51.85 (σ = 12.47)	40.21 (σ =8.83)	37.22 (σ = 7.56)
Accuracy (Word)	0.73 (σ = 0.12)	0.69 (σ = 0.15)	0.76 (σ = 0.12)	0.57 (σ = 0.15)
Accuracy (Textbase)	0.73 (σ = 0.12)	0.69 (σ = 0.18)	0.76 (σ = 0.12)	0.58 (σ = 0.12)
Accuracy (Situation)	0.73 (σ = 0.11)	0.66 (σ = 0.09)	0.76 (σ = 0.09)	0.64 (σ = 0.10)
Accuracy (Rhetorical)	0.73 (σ = 0.09)	0.72 (σ = 0.11)	0.73 (σ = 0.08)	0.64 (σ = 0.09)

4.2 Hierarchical Cluster Analysis

In addition to k-means clustering, we performed hierarchical cluster analysis, since k-means clustering is sensitive to the initial centroids and also does not do well with clusters with non-spherical shape and different size [16]. Hierarchical clustering is different from k-means clustering that directly divides a dataset into a number of disjoint groups. Hierarchical clustering proceeds successively either by merging smaller clusters into larger ones (bottom-up), or by splitting larger clusters into smaller clusters (top-down) [17]. We performed hierarchical clustering using Ward's method [26], and compared 4-cluster solution with 3-cluster and 5-cluster solution. Similar to what we found with k-means clustering, 4-clustering solution was most meaningful.

With the help of an R package clVaid [4], we compared the 4-cluster solution based on k-means clustering algorithm with the 4-cluster solution based on hierarchical clustering algorithm. The scores of the two solutions on three measures were computed. The measures were connectivity, Silhouette Width, and Dunn Index. Connectivity measures the degree of connectedness of the clusters based on the k-nearest neighbors, and the better solution minimizes it. The connectivity scores for k-means and hierarchical solutions were 118.69 and 13.31. Silhouette Width and the Dunn Index measure compactness and separation of the clusters. A higher Silhouette value indicates higher degrees of confidence in a clustering solution and a higher Dunn score indicates a better separated clustering solution. The Silhouette value for k-means and hierarchical solutions were 0.17 and 0.33, and the Dunn scores for k-means and hierarchical clustering were 0.10 and 0.27. Hierarchical clustering outperformed k-means clustering on all three measures, so the final solution we selected was the 4-cluster solution based on hierarchical clustering algorithm. The response time and performance accuracy of the four clusters is shown in Figure 2.

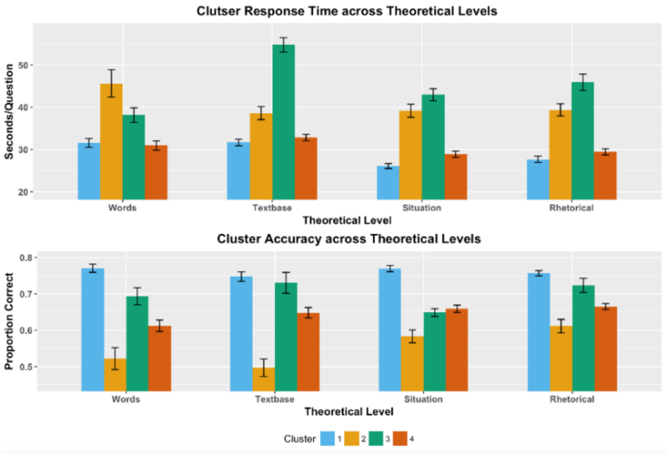


Figure 2: Time and accuracy of four clusters at four different theoretical level.

To compare the accuracy and time across clusters at different theoretical levels, we performed linear mixed-effects models using lme4 package in R [2]. We analyzed the effects of cluster and theoretical level on both proportion correct scores and time per question. In both models, subjects were specified as a random factor to control for the subject variance. For proportion correct scores, there was a statistically significant interaction between cluster and theoretical level, F (9, 747) = 4.38, p<.001, with the percentage of variance explained being 37.79%. For time per question, there was also a statistically significant interaction between cluster and theoretical level, F (9, 747) = 11.45, p<.001, vairance explained = 58.35%. However, time per question should be interpreted with caution since Situation Model and Rhetorical Structure lessons have multiple questions per text, which would shorten the expected time per question as learners had already built up their mental model for the text for most of the questions. Given these interactions, we will discuss the patterns of each cluster separately.

Cluster 1: Proficient readers

Cluster 1 is the biggest cluster with 39% (n = 98) of the study sample. These learners can be distinguished by their high speed and accuracy. As indicated by the results of mixed-effects models, the response time of Cluster 1 was shorter than the other three clusters at Situation Model and Rhetorical Structure level. At Word and Textbase level, there was no significant difference between the response time of Cluster 1 and Cluster 4, and Cluster 1 was faster than Cluster 2 and Cluster 3. Meanwhile, Cluster 1 achieved the highest proportion correct scores across all theoretical levels. Due to the students' high accuracy and short response time, we named this cluster "proficient readers." Proficient readers did not seem to be affected by theoretical level for accuracy, since they did equally well in lessons across different levels.

Cluster 2: Struggling readers

Cluster 2 is a smaller cluster with 12% of the study sample (n = 31). The response time of the learners in this cluster was comparatively long, and their accuracy was lower than the other clusters. According to the results of mixed-effects models, the response time of Cluster 2 on Word level questions was the longest, but their accuracy was the lowest. For Textbase, Situation Model and Rhetorical Structure level questions, the response time of Cluster 2 was the second longest, yet their accuracy remained the lowest among the four clusters. Due to the poor performance and long response time, we called this cluster "struggling readers." Unlike proficient readers who had stable performance across different theoretical levels, struggling readers did better in Situation Model and Rhetorical Structure lessons than Word and Textbase lessons.

Cluster 3: Conscientious readers

Cluster 3, like Cluster 2, contains 12% of the study sample (n = 31). The learners in Cluster 3 worked slowly and they achieved comparatively high performance accuracy. At Textbase, Situation model and Rhetorical Structure levels, the response time of Cluster 3 was the longest among the four clusters. Only at the Word level was the response time of Cluster 3 the second longest, trailing Cluster 2. Contrary to struggling readers who also worked slowly, Cluster 3 had the second highest accuracy. The proportion correct score differences between Cluster 1 and Cluster 3 ranged between 0.02 and 0.08 at different levels. We named this cluster "conscientious readers" because they achieved comparatively high accuracy through more effort. Similar to struggling readers, the conscientious readers' performance was associated with theoretical level. The results of mixed-effects models indicated that their performance at Textbase level was better than other levels.

Cluster 4: Disengaged readers

Cluster 4 is another large group representing 36% (n = 93) of the study sample. The learners in this cluster were almost as fast as the proficient readers, but their accuracy was comparatively low among the four groups. In particular, Cluster 4 learners were less accurate than both proficient readers and conscientious readers. The response time of Cluster 4 was as short as Cluster 1 at Word, Situation Model and Rhetorical Structure level. At Textbase level, the response time of Cluster 4 was the second shortest. However, there was a large gap between the performance of Cluster 4 and Cluster 1. Results of mixed-effects models indicated learners in Cluster 1 and Cluster 4 differed in their proportion correct score, and this difference ranged between 0.09 to 0.16, depending on the theoretical level. We named learners in Cluster 4 "disengaged readers" because of their short response time and comparatively poor performance. Theoretical level also affected disengaged readers, since they performed worse on Word level lessons than Textbase, Situational Model and Rhetorical Structure level lessons.

5. DISCUSSION

We developed AutoTutor for CSAL to teach adults with low literacy skills reading comprehension strategies in 35 lessons [9]. The lessons align with one or more of the following levels specified in Graesser-McNamara theoretical framework of comprehension [13]: Word, Textbase, Situational Model and Rhetorical Structure. To better understand how low literacy adult students interact with AutoTutor, we analyzed the online learning log of 253 adult students who participated in three intervention studies. Our first goal was to classify adult learners' behavior patterns while they interacted with AutoTutor. Our second goal was to investigate whether adult learners' behaviors were associated with different reading components represented by the theoretical levels.

Regarding the first goal, we identified four clusters of adult learners with distinctive learning behaviors through cluster analysis. We named the four clusters proficient readers, struggling readers, conscientious readers and disengaged readers. Proficient readers worked fast and accurately. Among the four clusters, the response time of the proficient readers was the shortest, meanwhile, their accuracy was the highest at the four theoretical levels. Opposite to proficient readers, struggling readers worked slowly and inaccurately. Their response time was either the longest or the second longest at different theoretical levels, however, their accuracy remained the lowest overall. Conscientious readers also worked slowly, but unlike struggling readers, they achieved comparatively high accuracy. The response time of conscientious varied across the theoretical levels, but they achieved similar high accuracy at all the theoretical levels. This indicated their awareness of their skill level versus effort needed for mastery. Similar to proficient readers, disengaged readers worked fast. However, their performance was not as good. These readers might try to get through lessons quickly without paying much attention to the content.

With respect to the second goal, we found learning behaviors of individuals in the four clusters varied across theoretical levels in different ways. Proficient readers performed equally well at different theoretical levels, but they spent less time on Situation Model and Rhetorical Structure level questions. One possible explanation for the variation in time across theoretical levels is that Situation Model and Rhetorical Structure lessons have many questions in each text. This could shorten the expected time per question as learners built their mental models after the first a few questions. Struggling readers' behaviors indicated an obvious effect of theoretical level. Struggling readers performed worse on questions addressing Word and Textbase levels than Situation Model and Rhetorical Structure levels. Although struggling readers' performance was poor, they were comparatively better at lessons with discourse components. For conscientious readers, their behavior on Textbase level lessons stood out. These readers spent more time on Textbase level questions, and as a result, they achieved higher accuracy on these questions than for questions addressing other theoretical levels. The behavior of disengaged readers varied the most when data for questions that tapped basic reading components and those questions concerning discourse components. Despite spending a similar amount of time on questions addressing Word and discourse levels, disengaged readers performance was better for discourse level material.

According to previous research [5, 24, 25], Word items place lower loads on working memory than discourse items. We thus assumed discourse level items would be more difficult than word level items, leading to better performance for the latter item type. Yet our data indicated that this assumption did not apply to the adult readers. Among the four types of readers we identified, the behavior of proficient readers and conscientious readers was not affected by whether the items tapped basic reading level or the discourse level processes. We considered that disengaged readers and struggling readers might be influenced by the distinction in item type (basic versus discourse), but the trend in our data actually showed the opposite, with higher accuracy of discourse level items than word level items. In addition to finding that behavior differed across clusters when comparing word to discourse levels, we also found that behavior varied between the three discourse levels. For example, the performance of conscientious readers was best for Textbase level items, but the performance of struggling readers was best for Rhetorical Structure level items. Our finding that learner behavior varies by discourse level suggests these levels represent distinguishable components of comprehension, and supports previous work on AutoTutor, which found the three discourse levels were separable since they were not highly correlated [14].

Based on the findings of this study, we suggest that clustering methods can be used to enhance the adaptivity of ITS. In particular, assessments and feedback can be personalized to assist different groups of students that exhibit particular patterns of learning behaviors. Differences in time and accuracy on theoretical levels indicate that ITS implementations that provide feedback on accuracy alone or on time alone would be misguided. Feedback and assessment in ITS that take into account both student trends in accuracy and time and their interaction, or lack of interaction, with theoretical level should better target the student type and prove to be more appropriate.

Apart from separating learners according to their distinct behavior patterns, we could also identify the learners' strength and weakness with regards to specific types of learning material. For example, some readers may struggle with word level but excel at discourse level comprehension. These readers might benefit more if the instruction is tailored towards word level comprehension training. Feedback based on a generalization that contains only one of these levels may be ineffective and miss groups of students entirely.

6. ACKNOWLEDGMENTS

This research was supported by the National Center of Education Research (NCER) in the Institute of Education Sciences (IES) (R305C120001) and the National Science Foundation Data Infrastructure Building Blocks program under Grant No. (ACI-1443068). This work is partially supported by the National Science Foundation Data Infrastructure Building Blocks program under Grant No. (ACI-1443068).

7. REFERENCES

[1] Baker, R. S., Corbett, A. T., Koedinger, K. R., & Wagner, A. Z. (2004, April). Off-task behavior in the cognitive tutor classroom: when students game the system. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 383-390). ACM.

[2] . Bates, D., Mächler, M., Bolker, B., & Walker, S. (2014). lme4: Linear mixed-effects models using Eigen and S4. *R package version, 1*(7), 1-23

[3] Bliuc, A. M., Ellis, R., Goodyear, P., & Piggott, L. (2010). Learning through face-to-face and online discussions: Associations between students' conceptions, approaches and academic performance in political science. *British Journal of Educational Technology, 41*(3), 512-524.

[4] Brock, G., Pihur, V., Datta, S., & Datta, S. (2011). clValid, an R package for cluster validation*. Journal of Statistical Software 25* (4).

[5] Cain, K. (2010). *Reading development and difficulties.* Oxford: Wiley-Blackwell.

[6] Craig, S. D., Gholson, B., Brittingham, J. K., Williams, J. L., & Shubeck, K. T. (2012). Promoting vicarious learning of physics using deep questions with explanations. *Computers & Education*, *58*(4), 1042-1048.

[7] Del Valle, R., & Duffy, T. M. (2007). Online learning: Learner characteristics and their approaches to managing learning. *Instructional Science*, *37*(2), 129-149.

[8] Fang, Y., Nye, B. D., Pavlik, P. I., Xu, Y. J, Graesser, A. C., & Hu, X. (2017, June). *Online learning persistence and academic achievement.* In Hu, X., Barnes, T., Hershkovitz, A., & Paquette, L. (Eds.), Proceedings of the 10th International Conference on Educational Data Mining (pp. 312-317). Wuhan, China: International Educational Data Mining Society.

[9] Graesser, A.C., Cai, Z., Baer, W.O., Olney, A.M., Hu, X., Reed, M., & Greenberg, D. (2016). Reading comprehension lessons in AutoTutor for the Center for the Study of Adult Literacy. In S.A. Crossley and D.S. McNamara (Eds.). *Adaptive educational technologies for literacy instruction* (pp. 288-293). New York: Taylor & Francis Routledge.

[10] Graesser, A. C., Forsyth, C. M., & Lehman, B. A. (2017). Two heads may be better than one: learning from computer agents in conversational trialogues. *Teachers College Record*, *119*(3), 1-20.

[11] Graesser, A. C., Hu, X., & McNamara, D. S. (2005). Computerized learning environments that incorporate research in discourse psychology, cognitive science, and computational linguistics. *Experimental Cognitive Psychology and its Applications: Festschrift in Honor of Lyle Bourne, Walter Kintsch, and Thomas Landauer. Washington, DC: American Psychological Association.*

[12] Graesser, A. C., Li, H., & Forsyth, C. (2014). Learning by communicating in natural language with conversational agents. *Current Directions in Psychological Science*, *23*(5), 374-380.

[13] Graesser, A. C., & McNamara, D. S. (2011). Computational analyses of multilevel discourse comprehension. *Topics in cognitive science*, *3*(2), 371-398.

[14] Greenberg, D., Graesser, A.C., Frijters, J.C., Lippert, A.M., & Talwar, A. (submitted). Using AutoTutor to track performance and engagement in a reading comprehension intervention for adult literacy students. Learning Disability Quarterly.

[15] Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics), 28*(1), 100-108.

[16] Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern recognition letters, 31*(8), 651-666.

[17] Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR), 31*(3), 264-323.

[18] Kalyuga, S., Ayres, P., Chandler, P., & Sweller, J. (2003). The expertise reversal effect. *Educational psychologist, 38(1)*, 23-31.

[19] Levin, J., Fox, J. & Forde, D. (2010). *Elementary statistics in social research*. Boston: Allyn & Bacon Pearson.

[20] Millis, K., Forsyth, C., Butler, H., Wallace, P., Graesser, A. C., & Halpern, D. (2011). Operation ARIES! A Serious game for teaching scientific inquiry. In M. Ma, A. Oikonomou, & J. Lakhmi (Eds.), *Serious Games and Edutainment Applications* (pp.169-196). London: Springer-Verlag.

[21] Mills, C., Graesser, A., Risko, E. F., & D'Mello, S. K. (2017). Cognitive coupling during reading. *Journal of Experimental Psychology: General*, *146*(6), 872-908.

[22] Nye, B. D., Graesser, A. C., & Hu, X. (2014). AutoTutor and family: A review of 17 years of natural language tutoring. *International Journal of Artificial Intelligence in Education*, *24*(4), 427-469.

[23] Pavlik Jr, P. I., Brawner, K., Olney, A., & Mitrovic, A. (2013). Tutoring Systems. *Design Recommendations for Intelligent Tutoring Systems: Volume 1-Learner Modeling, 1*, 39. US Army Research Laboratory.

[24] Perfetti, C.A. (2007). Reading ability: Lexical quality to comprehension. *Scientific Studies of Reading, 11*, 357-383.

[25] Van den Broek, P. W., White, M. J., Kendeou, P., & Carlson, S. (2009). Reading between the lines. Developmental and individual differences in cognitive processes in reading comprehension. In R. K. Wagner, C. Schatschneider & C. Phythian-Sence (Eds.), *Beyond decoding. The behavioral and biological foundations of reading comprehension* (pp. 107-123). New York, NY: The Guilford Press.

[26] Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of* the American Statistical Association, 58(301), 236-244.

[27] Wise, A. F., Speer, J., Marbouti, F., & Hsiao, Y. T. (2013). Broadening the notion of participation in online discussions: examining patterns in learners' online listening behaviors. *Instructional Science*, *41*(2), 323-343.

[28] Woodcock, R. W., McGrew, K. S., & Mather, N. (2001). *Woodcock-Johnson tests of achievement.* Itasca, IL: Riverside Publishing.

Feature extraction for classifying students based on their academic performance

Agoritsa Polyzou
Computer Science & Engineering Department
University of Minnesota
Minneapolis, MN 55454, USA
polyz001@umn.edu

George Karypis
Computer Science & Engineering Department
University of Minnesota
Minneapolis, MN 55454, USA
karypis@umn.edu

ABSTRACT

Developing tools to support students and learning in a traditional or online setting is a significant task in today's educational environment. The initial steps towards enabling such technologies using machine learning techniques focused on predicting the student's performance in terms of the achieved grades. The disadvantage of these approaches is that they do not perform as well in predicting poor-performing students. The objective of our work is two-fold. First, in order to overcome this limitation, we explore if poorly performing students can be more accurately predicted by formulating the problem as binary classification. Second, in order to gain insights as to which are the factors that can lead to poor performance, we engineered a number of human-interpretable features that quantify these factors. These features were derived from the students' grades from the University of Minnesota, an undergraduate public institution. Based on these features, we perform a study to identify different student groups of interest, while at the same time, identify their importance.

Keywords

academic student success, classification, feature importance

1. INTRODUCTION

Higher educational institutions constantly try to improve the retention and success of their enrolled students. According to the US National Center for Education Statistics [8], 60% of undergraduate students on four-year degrees will not graduate at the same institution where they started within the first six years. At the same time, 30% of college freshmen drop out after their first year of college. As a result, colleges look for ways to serve students more efficiently and effectively. This is where data mining is introduced to provide some solutions to these problems. Educational data mining and learning analytics have been developed to provide tools for supporting the learning process, like monitor and measure student progress, but also, predict success or guide intervention strategies.

Most of the existing approaches focus on identifying students at risk who could benefit from further assistance in order to successfully complete a course or activity. A fundamental task in this process is to actually predict the student's performance in terms of grades. While reasonable prediction accuracy has been achieved [14, 10], there is a significant weakness of the models proposed to identify the poor-performing students [18]. Usually, these models tend to be over-optimistic for the performance of students, as the majority of the students do well, or have satisfactory enough performance.

In this paper, we investigate the problem of predicting the performance of a student in the end of the semester before he/she actually takes the course. In order to focus on the poor-performing students, who are the ones that need these systems the most, the prediction problem is formulated as a classification task, where two groups of students are formed according to their course performance. We essentially identify two complementary groups of students, the ones that are likely to successfully complete a course or activity, and the ones that seem to struggle. After identifying the latter group, we can provide additional resources and support to enhance their likelihood of success.

However, "success" and "failure" can be relative or not. For example, a B− grade might be considered a bad grade for an excellent student, while being a good grade for a very weak student. We investigated different ways to define groups of students taking a course: failing students, students dropping the class, students performing worse than expected and students performing worse than expected, while taking into consideration the difficulty of a course.

In order to gain more insight into the learning process and its most important characteristics, we have created features that capture possible factors that influence the grades at the end of the semester. Using these features, we present a comprehensive study to answer the following questions: which features are good indicators of a student's performance? which features are the most important? The findings are interesting, as different features are the most important for different classification tasks.

The rest of the paper is organized as follows. Section 2 reviews the work in the area of predicting student performance in the end of the semester. In Section 3, there is an overview of the data that we used. Section 4 describes the features extracted, and Section 5 the classification tasks and methods tested. In Section 6, there is a detailed discussion of the experimental evaluation of the different methods tested, as well as the feature importance study. Section 7 contains the conclusions of the study.

2. RELATED WORK

As we are interested in estimating next-term student performance, we will review the related work in this area of research. The binary classification has been used in various educational problems, like predicting if a student will drop out from high school [6] or to predict if a student will pass a module in a distance learning setting [7]. Multi-label classification has been applied to provide a qualitative measure of students' performance. In [17], decision tree and naive Bayes classifiers are used with data from a survey. Attributes collected by a learning management system have been employed to estimate the outcome as Fail, Pass, Good and Excellent [16], or to classify students [12]. Some approaches [11, 9] test different ways to label the student performance, with two (pass or fail) or more labels. The majority of the aforementioned approaches are small-scale studies, that are applied to a limited number of courses.

In recent years, influenced by advances in the recommender systems, big data approaches have been also utilized in the area of learning analytics. Initially, the term "next-term grade prediction" was introduced by Sweeney et al. [18] in the context of higher education, and it refers to the problem of predicting the grades for each student in the courses that he/she will take during the next semester. Models based on SVD and factorization machines (FM) were tested. In another approach [15], the previous performance of students controls the grade estimation in two different ways while building latent models. In [19], some additional state-of-the-art methods were used, as well as, a hybrid of FM and random forests (RF). The data used are the historical grades and additional content features, representing student, course and instructor characteristics. At the same setting, [14] and [10] developed course-specific methods to perform next-term grade prediction based on linear regression and matrix factorization.

All these methods assign a specific numerical grade to each student's attempt to take a course. A limitation identified in these approaches was that the developed models perform poorly for failing students. In [5], failing students have been completely removed from the dataset. As this is the subpopulation of students that needs additional support the most, it is very important for a model to be able to accurately identify these students at risk.

This work is a more general study of the factors that influence the student performance, in a very large scale. The only observed data that we have available are the students' grades at the end of the semester. In our approach, we formulated this problem as a binary classification task, in order to detect the different group of students. In other words, we keep the classification methodology, but apply it on the context of big data.

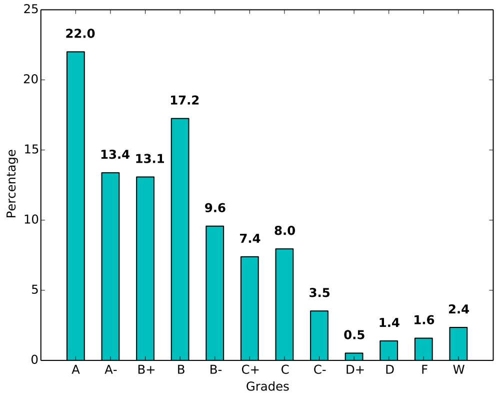


Figure 1: Percentage of each letter grade with respect to the total grades.

3. DATASET

First, we will clarify the use of some terms in the current context. An *instance* refers to the performance of a student, s, in a course, c, at the end of the semester. All the courses that a student took in past semesters, before taking course c, are the *prior courses*, denoted by $C_{s,all_{prior}}$. The set of courses for a single semester x is denoted as $C_{s,x}$. Additionally, for a course c there might exist a stated set of courses that are required for a student to take before attempting c. We refer to this set as the *prerequisite courses*. Every course x worths a specified number of credits, cr_x.

An undergraduate student enrolled to a college or university has to take some courses each semester, and receive a satisfactory grade in order to successfully complete them. Depending on the student's degree program, these courses might be required, electives, or simply courses that the student takes for his/her own advancement, intellectual curiosity, or enjoyment. If a student withdraws from a course after the first two weeks of classes, it is denoted by the letter 'W' in the student's transcript.

The original dataset was obtained from the University of Minnesota and it spans over 13 years. We removed any instances with a letter grade not in the A–F grading scale (A, A−, B+, B, B−, C+, C, C−, D+, D, F). Statistics about the grades in the dataset are shown in Fig. 1. In our dataset, the letter grade A is the most common. We extract features for the instances occurring during the last 10 fall and spring semesters. Given a semester, we utilize all the students that had taken the course before, and for each student taking a course, we extract a set of features. Additionally, we generate features for the instances awarded with the letter W, but we do not utilize them in any other way during the feature extraction process. These will be used only when trying to predict the students that drop-out from a course.

4. EXTRACTED FEATURES

Having as input the historical grading data, we derived different features to capture possible factors for a student's poor performance. The features can be separated into three distinct categories: the student-specific (independent from course c), course-specific features (independent from student

s) and student- and course-specific features (they are a function of both s and c). All extracted features are described in Tables 1, 2, where related features are grouped together into eight different subcategories. The keywords on bold are used to indicate the corresponding group of features later. Note that for each $\{s, t, c\}$, where student s took course c in semester t, we generate a different set of features. Every set of features characterize a student's attempt to take course c at the specific point of his/her studies.

These features are either numerical, categorical or indicator variables. For indicator features, we use the values of 0 or 1. The categorical features are encoded via a numerical value. For example, the feature about the current semester is categorical, and the values {fall, spring, summer} are transformed to {0,1,2}, respectively.

5. CLASSIFICATION PROBLEMS

5.1 Classification tasks

Our motivation was to identify groups of students that need further assistance and guidance in order to successfully complete a course. These students could benefit from informed interventions. We consider this to be a binary classification problem, where these students form one of the classes and the remaining students form the other class.

We consider different ways of measuring when a student does not do well in a course to deal with the performance measurement challenges we mentioned earlier. Unsatisfactory performance can occur when the earned grade represents a performance that is bellow the student's potential. We considered the following four ways for labelling, resulting to these absolute and relative classification tasks:

1. Failing student performance, i.e., letter grades D and F (denoted as the **Fgr** task).
2. The letter grade W (denoted as the **Wgr** task). This represents the instances when the student dropped the course. This behavior is worrisome as it shows that either the student was not interested in the course anymore or he/she expects to perform poorly.
3. Student performance that is worse than expected, i.e., the grade achieved is more than two letter grades lower than the student's GPA (denoted as the **RelF** task).
4. Student performance that is worse than expected while taking into consideration the difficulty of the course (denoted as the **RelCF** task). The difficulty of a course is expressed by the average grade achieved by the students that took the course in prior offerings. A positive instance is when the grade achieved is more than two letter grades lower than the average of the student's GPA and the course's prior average grade.

Statistics for the different classification tasks can be found at Table 3.

As discussed at the related work section, it is easier to predict the successful students. In order to have a better understanding of the relative difficulty of this task compared with the four tasks mentioned above, we also examined the task of predicting the students that completed a course with the grade A (denoted as the **Agr** task).

5.2 Methods compared

In order to support students that need help to successfully complete a course, we will use classification techniques to identify them from the rest of the students. The instances of interest will be labeled as 1, and the rest as 0. The problem can be described as follows. We are given a set of training examples that are in the form $(\mathbf{x}, y)$ and we want to learn their structure. We assume that there is some unknown function $y = f(\mathbf{x})$, that corresponds the feature vector $\mathbf{x}$ to a value y. In our case, $y = \{0, 1\}$. A classifier is an hypothesis about the true function f. Given unseen values of $\mathbf{x}$, it predicts the corresponding y values.

We tested the following classifiers [4], using scikit-learn library in Python [13]: Decision Tree (DT) [2] and Linear Support Vector Machine (SVM) [3] as base classifiers, and Random Forest (RF) [1] and Gradient Boosting (GB) [4] as ensemble classifiers.

While using DT, the classification process is modeled as a series of hierarchical decisions on the features, forming a tree-like structure. In other words, we ask a series of questions about the features of an instance, and based on the answer, we may ask more questions, until we reach to a conclusion about the class label of that instance. The goal is to get a split that allow us to make a confident prediction. Consider the m-dimensional space that is defined by the feature vectors $\mathbf{x}$, of length m. There, every training instance corresponds to a single point. A Linear SVM looks for a decision boundary between two classes, a hyperplane that bisects the data with the largest possible margin between the two different classes. The margin on each side of the hyperplane is the area with no data points in it.

Ensemble methods try to increase the prediction accuracy by combining the results from multiple base classifiers. RF is a class of ensemble methods that uses decision trees as week learners. Randomness has been explicitly inserted in the model building process, as every splitting criterion considers only a subset of features, randomly selected from the feature vector of $\mathbf{x}$, to select the best split. Once we build all the trees, the majority class is reported. In boosting, a weight is associated with each training instance. Using the same algorithm, classifiers are training on a weighted training set to focus on hard-to-classify instances. At the end of each iteration, the weights of instances with high misclassification error are relatively increased for future iterations. In GB for binary classification, a single regression tree is built, where in each splitting criterion, only a subset of the features is considered. Once the tree is built, then, the corresponding weight of the classifier in the current iteration is estimated.

6. EXPERIMENTS

6.1 Experimental design

The models constructed are global, i.e., a single model predicts the performance of all students over all the courses. All features are extracted for any instance of a student taking a course. As randomization takes part in the models while sampling and/or initialization, we run the same model with

Table 1: Feature groups describing the target student s in the target semester t.

(1) Student's status in terms of grades. (**grades**)	• Average grade of s in prior courses C_{s,all_prior}. $\sum_j g_{s,j}/\lvert C_{s,all\_prior}\rvert$, for j in C_{s,all_prior}. • GPA of s, i.e., weighted average of the grades in prior courses w.r.t. the credits worth. $\sum_j g_{s,j}cr_j/\sum_j cr_j$, for j in C_{s,all_prior}. cr_j is the number of credits of course j. • GPA of s over the prior courses that belong in his/her major. • GPA of s over the prior courses that do not belong in his/her major. • GPA of s over the courses taken the previous semester, i.e. at the semester (t-1). • GPA of s over prior courses taken the past two semesters i.e. at semesters (t-1) and (t-2). • GPA of s over prior courses taken on fall, spring and summer semesters. Essentially, here there are 3 features, one for each semester type. • Average grade of courses that s took with the same corresponding credit. There are 6 different features, each corresponding to prior courses that worth 1,2,3,4,5 or 6 credits. • GPA of courses that s took at the same course level. There are 6 levels (1xxx, 2xxx, 3xxx, 4xxx, 5xxx, or 8xxx). Higher level courses are more advanced.
(2) Other info indicating a student's status. (**status**)	• The number of prior courses, $\lvert C_{s,all\_prior}\rvert$. • Student's major. Included majors: Aerospace Engineering, Biomedical Engineering, Chemical Engineering, Chemistry, Civil Engineering, Computer Science, Electrical Engineering, Materials Science, Mathematics, Mechanical Engineering, Physics, and Statistics. • The total credits that s has earned in prior courses. $\sum_j cr_j$, for j in C_{s,all_prior}. • Indicator whether target semester t is a fall, spring, or summer semester. • Indicator whether the student has ever registered for the summer semester. This is an indicator of the past behavior of the student. • The number of semesters that the student is active, $\text{nterms\_active}_{s,t}$. • The number of years student s is in the program. • The number of transferred credits. It is quite common for students to transfer some credits from other institutions, or from qualified courses they took at high school.
(3) Student's course load. (**load**)	• Average credits s earned in prior courses per semester. $\sum_j cr_j/\text{nterms\_active}_{s,t}$, for j in C_{s,all_prior}. • The number of credits s earned in the past semester. $\sum_j cr_j$, for j in $C_{s,t-1}$. • The number of credits earned in the current semester. $\sum_j cr_j$, for j in $C_{s,t}$. • The number of courses taken the current semester. $\lvert C_{s,t}\rvert$. • Ratio of s's course load in the current semester to his/her average course load over the past semesters. This is a way to compare the usual load of the student with the load for the target semester. $(\sum_i cr_i/(\sum_j cr_j/\text{nterms\_active}_{s,t})$, for i in $C_{s,t}$ and j in C_{s,all_prior}.

The set C_{s,all_prior} represents the courses that the student took all the prior semester, before the target semester t. For any semester x, $C_{s,x}$ represents the set of courses that student s took on semester x.

5 different seeds and average out the performance achieved. We used cross validation for classifier evaluation. The data are partitioned into 5 disjoint subsets. For each fold, test on one partition and use the remaining ones for training. The average of the evaluation metrics across the 5 folds will be the values reported.

Metrics. Precision is the ratio of true positives to all predicted positives. Recall is the ratio of true positives to all actual positives. Precision is intuitively the ability of the classifier not to label as positive a sample that is negative, while recall is the ability to find all the positive samples. F_1 score is a measure of accuracy, calculated as:

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{1}$$

Area under the receiver operating characteristic (ROC) curve, AUC, is also reported to understand the performance of a classifier w.r.t. all the thresholds. ROC curve plots the true positive rate against the false positive rate, at various thresholds. AUC corresponds to the probability that the classifier will rank a random positive instance higher than a negative one.

Estimating positive threshold. Instead of assigning a label to a test instance, we can assign a prediction score in the range of [0,1] that will be the probability of the input samples to belong to the positive class. In this way, we will be able to compute metrics like AUC. To estimate a threshold of the prediction score above which the object is assigned to the positive class, we follow these steps: 1. Sort the prediction scores in non-increasing order. 2. For each point L in this sorted sequence, compute the F_1 score, using Eq. 1, by assuming that any instances that have a prediction score that is greater than that of the Lth instance is classified as positive and everything else is classified as negative. 3. The F_1 score is the maximum F_1 value obtained above.

Table 2: Feature groups describing the student s in term t and course c.

(4) Course's difficulty and popularity. (**c-diff**)	• Relative course load when s took c w.r.t. the average credits of past students at the semester they had taken c. For each past student, compute the number of credits earned on that semester. Then, compute the fraction of $\sum_j cr_j$, for j in $C_{s,t}$, divided by the average credits earned from past students on the same semester that they took course c. Values greater than 1 indicate heavier load than other students. • Average grade earned by past students. • Average grade in c of past students within the same major as the s. Now, filter the students in order to keep only the students that are in the same department as s. • Average grade in c of students belonging to c's major or not. This describes two features, by separating the past students to the ones that are in the same major as the department of c, and the ones that are out-of-the-department.
(5) Performance / Familiarity with the course's background and department. (**c-backgr**)	• Fraction of students in the same major as s that have taken the c. This feature measures how popular is course c across the students on the department of student s. • Fraction of students from s's major that took c, shows how common is c in s's major. • Number of courses that s took and belong to c's department. Absolute measurement of how familiar is s with the department of the course c. • Ratio of courses that s took and belong to c's department. Relative measure of how familiar is s with the department of the course c. • Ratio of credits that s took and belong to c's department. Relative measurement of how familiar is s with the department of the course c, in terms of credits. • Ratio of credits that s took and belong to c's department and the average credits that past students took and belonged to c's department. This is a relative measurement of how familiar is s with the department of the course c, in comparison with past students. • GPA over the courses that s took and belong to c's department. This feature is a quantitative measure of student's performance in the c's department.
(6) Information about the prerequisites. (**prerequ**)	• GPA of the prerequisite and non-prerequisite courses that s has taken. Two features that show the performance of the student in prerequisite and other courses. • Number of the prerequisite courses taken by s, an absolute measurement. • Ratio of prerequisite courses taken by s. Relative measure to show how much well-prepared the student is, in terms of the stated prerequisites. • Average terms past since prerequisite courses were taken by s.
(7) Performance relative to the course's level. (**c-perform**)	• The number of lower, same and higher level courses w.r.t. the level of c. • GPA over lower, same, higher level courses w.r.t. the level of c.
(8) Course-specific features. (**c-spec**)	• Course level that c belongs to. • Indicator whether c is in the student's major or not. • Average grade earned by past students.

The set C_{s,all_prior} represents the courses that the student took all the prior semester, before the target semester t. For any semester x, $C_{s,x}$ represents the set of courses that student s took on semester x.

Table 3: Statistics for the different classification tasks.

Task	Fgr	Wgr	RelF	RelCF	Agr
# instances	94,364	96,941	94,364	94,364	94,364
# positive	3,139	2,577	20,398	21,724	20,851
% positive	3.33	2.7	21.62	23.02	22.10

6.2 Performance analysis

Table 4 summarizes the performance of the various classification methods for the classification tasks, in terms of the AUC and the F_1 score. Based on both metrics, GB is the best performing method, closely followed by the RF classifier. As expected, DT, which is the simplest method, has the lowest performance. These results are better compared to the performance of grade prediction methods for any classification task. When using Course-Specific Regression for predicting the failing students, we get a F_1 score of 0.118, which is lower than any of the other methods we discuss.

While comparing the classification tasks, we can see that the tasks that predict relative performance have lower AUC values than when predicting absolute performance. In terms of F_1 scores, we can see clearly that the A-students are the most accurately predicted. The F_1 scores of the different tasks are related to the percentage of positive instances in each task. The tasks Fgr and Wgr, that are highly unbalanced, have significantly lower F_1 scores. Moreover, as there is 81% overlap between the students that are positive for both RelF and RelCF, the tasks of RelF and RelCF have very similar performance.

6.3 Feature importance study

One of our goals is to study which factors are important indicators of a student's performance, so we performed the

Table 4: Performance of the various classifiers.

Area under the ROC curve					
Classiffier	Fgr	Wgr	RelF	RelCF	Agr
DT	0.834	0.710	0.689	0.716	0.820
SVM	0.853	0.736	0.690	0.718	0.819
RF	0.873	0.778	0.748	0.759	0.850
GB	**0.877**	**0.780**	**0.755**	**0.765**	**0.854**

F_1 score					
Classifier	Fgr	Wgr	RelF	RelCF	Agr
DT	0.255	0.123	0.450	0.466	0.573
SVM	0.276	0.171	0.452	0.469	0.570
RF	0.317	0.165	0.499	0.502	0.604
GB	**0.319**	**0.181**	**0.506**	**0.507**	**0.610**

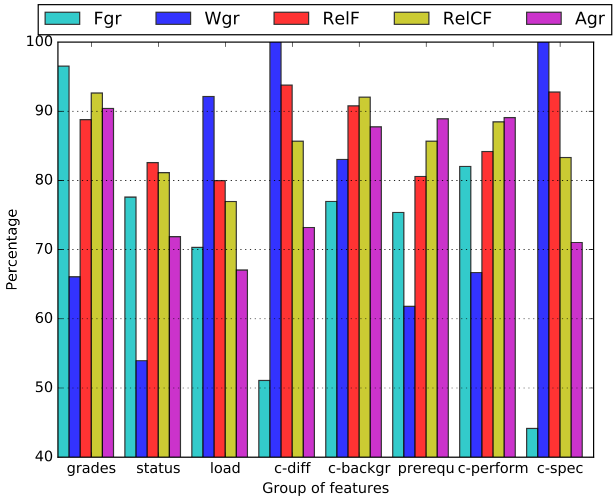


Figure 2: Percentage of performance managed to recover using only one group of features.

following experiment. We categorize each extracted feature to one of the the 8 groups, according to Table 1. Afterwards, for each classification task, we built RF classifiers using only the features belonging to one of the above groups. We selected to use RF over GB, as they achieve similar performance in less training time. The accuracy achieved for a model using a single group of features is expected to be less than the accuracy when using all the features. The percentage of accuracy that a model using only the features belonging to one group manages to achieve, in terms of the F_1 score, are presented on Fig. 2. In this bar chart, we can see the percentage of accuracy achieved from all the different feature groups for all the discussed classification tasks. The higher the percentage achieved by a single group of features, the more predictive ability these features have.

From this figure, we can get many insights on the factors that affect student performance. For example, the features related to the students' grades (group 1) have a very good predictive capability in almost all the tasks, except the task of predicting the W grades. In this task, features related with the course's difficulty and popularity (group 4) as well as features that are course-specific (group 8), manage to achieve the same accuracy as when using all the features. This indicates that the reasons that a student drops a course are related more to the course, rather than to the students themselves. The next best indicator is the feature group about the student's course load during the semester.

On the other hand, this is not the case for predicting the failing students, in the absolute sense, i.e., receive a D or F. When using only course-related groups (groups 4, 8) for predicting the students likely to fail a course (Fgr task), we manage to recover half or less from the F_1 score. As a result, these factors do not influence the absolute failing performance of a student, indicating that the reasons for that are mostly related with the student. As the students' grades manage to recover almost the same performance as when using all the features, they are the ones that affect the Fgr prediction the most. When using the other groups, it is very difficult to achieve comparable performance, as they recover 80% or less of the F_1 score.

The feature groups are behaving similarly for RelF and RelCF. However, we notice that for the RelCF task, the feature groups that are related with student-course specific features have slightly better performance, while the student-specific groups have slightly worst performance, compared to the task of RelF. This is happening because, for RelCF, we take into consideration how other students usually perform on the target course. Every single group has enough information for the RF to utilize to achieve performance which is as good as 75% of the best case, i.e., when using all the features.

Finally, for identifying the A-students, the feature groups 1, 5, 6, 7 are the ones that manage to have the best performance. These groups are related with students' grades in general, but also, with their grades relative to the target course's background, prerequisites and level. Using only one of them can provide us with the information we need in order to recover around 90% of the performance while using all the features.

7. CONCLUSIONS

The purpose of this paper is to accurate identify students that are at risk. These students might fail the class, drop it, or perform worst than they usually do. We extracted features from historical grading data, in order to test different simple and sophisticated classification methods based on big data approaches. The best performing methods are the Gradient Boosting and Random Forest classifiers, based on AUC and F_1 score metrics. We also got interesting findings that can explain the student performance.

8. ACKNOWLEDGEMENTS

This work was supported in part by NSF (IIS-1247632, IIP-1414153, IIS-1447788, IIS-1704074, CNS-1757916), Army Research Office (W911NF-14-1-0316), Intel Software and Services Group, and the Digital Technology Center at the University of Minnesota. Access to research and computing facilities was provided by the Digital Technology Center and the Minnesota Supercomputing Institute.

9. REFERENCES

[1] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[2] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[3] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[4] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[5] Z. Iqbal, J. Qadir, A. N. Mian, and F. Kamiran. Machine learning based student grade prediction: A case study. *arXiv preprint arXiv:1708.08744*, 2017.

[6] J. E. Knowles. Of needles and haystacks: Building an accurate statewide dropout early warning system in wisconsin. *Journal of Educational Data Mining*, 7(3):18–67, 2015.

[7] S. Kotsiantis, C. Pierrakeas, and P. Pintelas. Predicting students'performance in distance learning using machine learning techniques. *Applied Artificial Intelligence*, 18(5):411–426, 2004.

[8] J. McFarland, B. Hussar, C. de Brey, T. Snyder, X. Wang, S. Wilkinson-Flicker, S. Gebrekristos, J. Zhang, A. Rathbun, A. Barmer, et al. Undergraduate retention and graduation rates. In *The Condition of Education 2017. NCES 2017-144*. ERIC, 2017.

[9] B. Minaei-Bidgoli, D. A. Kashy, G. Kortemeyer, and W. F. Punch. 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, pages T2A–13. IEEE, 2003.

[10] S. Morsy and G. Karypis. Cumulative knowledge-based regression models for next-term grade prediction. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 552–560. SIAM, 2017.

[11] E. Osmanbegović and M. Suljić. Data mining approach for predicting student performance. *Economic Review*, 10(1):3–12, 2012.

[12] A. Pardo, N. Mirriahi, R. Martinez-Maldonado, J. Jovanovic, S. Dawson, and D. Gašević. Generating actionable predictive models of academic performance. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 474–478. ACM, 2016.

[13] 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. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[14] A. Polyzou and G. Karypis. Grade prediction with models specific to students and courses. *International Journal of Data Science and Analytics*, 2(3-4):159–171, 2016.

[15] Z. Ren, X. Ning, and H. Rangwala. Grade prediction with temporal course-wise influence. In *Proceedings of the 10th International Conference on Educational Data Mining*, pages 48–55, 2017.

[16] C. Romero, S. Ventura, P. G. Espejo, and C. Hervás. Data mining algorithms to classify students. In *Educational Data Mining 2008*, 2008.

[17] A. A. Saa. Educational data mining & students' performance prediction. *International Journal of Advanced Computer Science & Applications*, 1:212–220, 2016.

[18] M. Sweeney, J. Lester, and H. Rangwala. Next-term student grade prediction. In *Big Data (Big Data), 2015 IEEE International Conference on*, pages 970–975. IEEE, 2015.

[19] M. Sweeney, H. Rangwala, J. Lester, and A. Johri. Next-term student performance prediction: A recommender systems approach. *arXiv preprint arXiv:1604.01840*, 2016.

Identifying User Engagement Patterns in an Online Video Discussion Platform

Seung Yeon Lee
EdLab, Teachers College
Columbia University
tmddus30@gmail.com

Hui Soo Chae
EdLab, Teachers College
Columbia University
hsc2001@tc.columbia.edu

Gary Natriello
EdLab, Teachers College
Columbia University
gjn6@columbia.edu

ABSTRACT

In this study we conducted behavioral analyses to gain insights into patterns of user interaction in a video discussion platform, *Vialogues*. Vialogues provides an asynchronous online discussion environment around video. Using a hierarchical clustering analysis on users' clickstream data, we identified four different behavior patterns: (1) video watchers with no discussion activity, (2) opinion seekers and active repliers with little to no video watching activity, (3) users who watched and discussed videos, and (4) users focused on viewing and/or creating metadata. Despite being the largest group, Cluster (3) had the least classifiable characteristics. Consequently we conducted additional analyses to examine finer-grained user segments. For each segment we created a transition network using weighted directed networks in order to understand the transition pattern between two consecutive click activities.

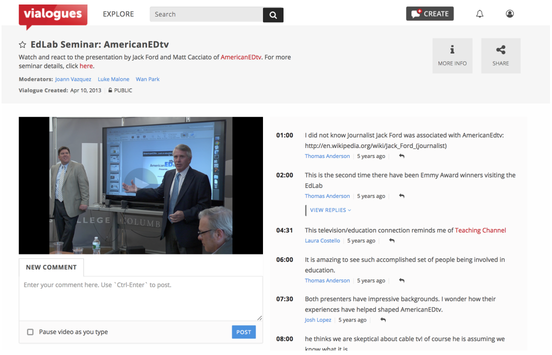
Keywords

Online discussion, video learning, hierarchical clustering, transition network, user behavior

1. INTRODUCTION

Using video as an instructional technology has been a popular approach across a broad range of educational contexts. Video provides a way for learners to immediately connect with subject matter; increasingly, it also provides a way for learners to connect with each other. A number of benefits to using video in education have been reported over several decades of research [7, 8, 10, 11, 5]. While traditional video platforms primarily support passive learning, social video platforms, (e.g., YouTube), provide an active learning environment for learners to discuss video and share content collaboratively.

Vialogues [1] is an asynchronous online video discussion platform that facilitates collaborative conversations around video. The platform allows users to comment directly on specific points in time of a video, as opposed to commenting only in the comment section that references the entire video. The main video is shown on the left side of the screen, and the discussion board is shown on the right side of the screen, as shown in Figure 1. As described above, all comments are coded to a specific point in time in the video, and the related portions of the video are referenced. Also, the discussions are threaded so that users are able to view and respond to one another's comments. The addition of this feature in Vialogues allows deeper understanding of video by enabling users to understand the context and to discuss via conversation threads; this resolves one of the main problems of many existing video-discussion tools.



☆ EdLab Seminar: AmericanEDtv

Watch and react to the presentation by Jack Ford and Matt Cacciato of AmericanEDtv. For more seminar details, click here.

Moderators: Joann Vazquez Luke Malone Wan Park

Vialogue Created: Apr 10, 2013 | PUBLIC

NEW COMMENT

Enter your comment here. Use `Ctrl-Enter` to post.

□ Pause video as you type

01:00 I did not know Journalist Jack Ford was associated with AmericanEdtv: http://en.wikipedia.org/wiki/Jack_Ford_(journalist)
Thomas Anderson | 5 years ago

02:00 This is the second time there have been Emmy Award winners visiting the EdLab
Thomas Anderson | 5 years ago

04:31 This television/education connection reminds me of Teaching Channel
Laura Costello | 5 years ago

06:00 It is amazing to see such accomplished set of people being involved in education.
Thomas Anderson | 5 years ago

07:30 Both presenters have impressive backgrounds. I wonder how their experiences have helped shaped AmericanEDtv.
Josh Lopez | 5 years ago

08:00 he thinks we are skeptical about cable tv! of course he is assuming we know what it is

Figure 1: An example vialogue page. The page shows the title and description of the vialogue (top), the video player (left) and discussion panel (right).

Vialogues provides a comprehensive set of pedagogical tools to assist teachers to flexibly design and monitor learning activities, as well as receive feedback from students based on instructional needs. Teachers can ask either survey questions, with a 'check all that apply' answer option or devise poll questions, with a 'single answer' option. Teachers are able to present these questions throughout different points in the video. In addition, users can either open their discussion to everyone, or they can restrict access to some specific users. With these additional tools, discussion moderators can effectively control the quality of their discussion and tailor a discussion for an intended group of users.

In this paper, we conducted behavioral analyses to gain insights into users' interaction patterns in Vialogues. We performed several types of analyses on users' clickstream data in a step wise way to develop deeper understanding of user

Table 1: Four event categories on Vialogues, (1) video player, (2) video watch, (3) discussion, (4) other features, and corresponding event actions of each event category

Video Player	Video Watch	Discussion			Other Features
		Comments	Reply	Poll	
Video ready	Watch 3 seconds	Post comments	Click reply	Post poll	Pause as typing
Play	Watch 10 seconds	Click edit comment	Reply	Add poll item	Click time code
Pause	Watch 30 seconds	Cancel edit comment	Expand reply	Remove poll item	Open Vialogues tab
Mute true	Watch 50%	Update comment	Hide reply		Close Vialogues tab
Mute false	Watch 95%	Click delete comment			Open settings tab
Fullscreen true	Watch 100%	Delete comment			Save edit Vialogue
Fullscreen false					Save edit Video
					Cancel edit Vialogue

engagement. First, we examined the overall usage patterns based on the distribution of different user actions. Then we investigated various interaction patterns by using a hierarchical clustering analysis. The clustering analysis identified four user groups and we conducted in-depth analysis on each groups to understand its distinctive behavior characteristics. We found that different groups demonstrated different levels of engagement, particularly in terms of discussion activity. The findings of this study could be used to create a useful reference for designing instruction based on video discussion tools or for developing this kind of learning platforms.

2. DATA

2.1 Data Source

We analyzed the clickstream actions generated as users interact with Vialogues. We considered four event categories to understand users' behaviors on a particular vialogue page (e.g., Figure 1). First, we collected users' actions interacting with the video player, e.g., play, pause, mute, full screen. Second, we recorded the video completion rate, e.g, watch the first 3 seconds, watched 50% of the total duration of video, etc. Third, we collected actions related to discussion. For context, Vialogues supports three different ways to participate in a discussion: commenting, replying and posting polls. Actions related to each of these three activities were collected, e.g, post comments, reply to others' comments, expand replies, post a poll, etc. Lastly, other actions were also tracked including whether they used a "Pause as Typing" feature which automatically pauses the video while typing comments; whether they clicked the "Time Code" in the discussion panel to find the corresponding part of video; and whether they opened the vialogues tab to see more information about the particular vialogue such as the uploader or the shareable link. Table 1 presents the full list of tracked actions under each of the four categories described above, which we defined as 'Video Player', 'Video Watch', 'Discussion (Comments, Reply, Poll)' and 'Other Features' respectively. We used the data of users who visited a vialogue page during the month of September 2017.

2.2 Data Pre-processing

We conducted data cleaning and exploratory analyses to preprocess the original sample data. First, we excluded vialogue contents created by Vialogues administrators. Second, we examined the number of different vialogue pages a user visited within a single session in order to understand its distribution and detect any outliers; this numbers varied from 1 to 37 but 95% of the data had values between 1 and 3. We only considered those 95% of the data and assumed that the data with values greater than 3 were outliers. We believe that when a user explores too many contents within the same session, he or she is less likely to be fully engaged in watching videos or participating in discussions. Next, we processed data to create a vector for the sequence of actions for each case, where cases were defined as a unique vialogue page within a session for each user. For example, if a user interacted with two different vialogue pages during a given session, two vectors were defined for this user. We treated them as separate cases because users' behaviors can not only be different by sessions but also by vialogue contents. From this process, we created 6,706 distinct cases from the September data. In other words, the total unique sets of sessions and vialogue pages viewed in September 2017 were 6,706. In vialogue pages, by the system setting, users' first action were logged as "Video Ready". However, 563 cases out of the total 6,706 started with different event actions other than video ready. This indicates that some users could interact with the same vialogue pages through multiple sessions. For example, users may pause their interaction with Vialogues and come back to the same page after some period of time. The event actions of those aforementioned sessions are dependent with the actions of the corresponding previous sessions. Thus, for the purpose of our analysis, these 563 cases were excluded as they could not be treated as independent cases like the others. We also examined the length of action sequences. We found that its distribution is extremely skewed: 90% of the cases were between 1 and 50 and there were outliers with extreme values up to 357. After deleting these outliers, we also excluded the first action, video ready, from every vector. This event action was created by default, not by users' intention, and is present in every cases, so does not provide any useful information about the user engagement. After such data processing steps, the analytic sample from September 2017 included 3,485 unique cases of 2,972 sessions from 1,516 unique user IDs and 991 unique vialogue contents.

3. OVERALL PATTERN OF USER INTERACTION

In order to understand the overall pattern of users' behavior on vialogue pages, we first evaluated the frequencies of event actions in September 2017. This analytic sample, constructed by the process described above, included 33 event actions from the actions given in Table 1, and excludes "Video Ready" as described above. Table 2 shows the frequency count of the top 10 most frequent actions. The distribution suggests that the actions of playing or pausing

Table 2: Frequency of the top 10 most frequent actions

Event Action	Freq
Video pause	9,818
Video play	8,445
Expand reply	4,262
Video watch 3 Secs	3,070
Video watch 10 Secs	2,839
Video watch 30 Secs	2,575
Post comment	2,157
Video watch 50%	1,878
Video watch 95%	1,181
Click reply comment	1,179
Video fullscreen true	1,109

videos occurred the most frequently, which was expected. The next most frequent action, interestingly, was expanding replies. In the platform, users can view others' first-level comments without any click activity but they need to click to read replies to a particular comment. Although the system cannot capture the action of simply reading others' comments without clickable actions, "Expand reply" can serve as a proxy for the behavior of browsing others' discussion comments. Then, the events "Watch for 3 seconds," "Watch for 10 seconds," and "Watch for 30 seconds" were the next most frequent actions in that order. The next most frequent action was "Post comment" followed by "Video watch 50%" then "Video watch 95%." This shows that users are not necessarily watching the complete video before commenting. It also suggests that some users are commenting while watching a video. One interesting finding regarding replying to comments is that the count of "Reply comment" (freq = 907) is less than the count of "Click reply comment" (freq = 1,179). It is possible that some users tried to reply to a comment but ultimately decided not to post a reply. It is also possible that some users were confused by the platform's layout and clicked the button by mistake. Overall, it appears that viewers typically browse discussion contents, watch a video for a certain duration, and post comments (not replies) before completing a video. It also shows that people reply to comments (freq = 907) less frequently than completing 95% of a video (freq = 1,181). Note that these observations are at the aggregate level and do not take into consideration different user sessions or time sequences. However, our goal is not generalization. Instead our goal is to develop a better sense of overall user activity patterns.

4. IDENTIFYING DISTINCT BEHAVIOR PATTERNS

4.1 Method

To further investigate user groups with distinct behavior patterns, we conducted a clustering analysis. For clustering, we created a $M \times N$ matrix, where M is the total number of cases and N is the number of different actions. Columns indicate unique actions and rows indicate cases defined as unique vialogue pages visited within an individual session. The elements of each row are the frequency counts of each action in each case. In the data processing step, we found large variation in the total number of actions among different cases, and this may cause invalid clustering results. To minimize such risks, we normalized each row and then computed the distances between the pair of rows using the cosine similarity. The cosine similarity measures the similarity based on the angle between two vectors ignoring the frequency of each element. For two vectors, $\boldsymbol{a} = \{a_i\}$ and $\boldsymbol{b} = \{b_i\}$ for $i = 1, ..., M$, the cosine similarity is calculated by $Similarity = cos(\theta) = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{||\boldsymbol{a}||_2 ||\boldsymbol{b}||_2} = \frac{\sum_{i=1}^{M} a_i b_i}{\sqrt{\sum_{i=1}^{M} a_i^2}\sqrt{\sum_{i=1}^{M} b_i^2}}$. The corresponding distance was computed by $1 - Similarity$. Using the cosine-based distances, we applied Ward's hierarchical cluster method [6]. Specifically, we used the "agglomerative approach" which goes from the bottom up. This approach starts with each data point in a cluster of its own. Then, it repeats the process of finding the most similar pair of clusters and merging them until all data are merged into only one cluster. The Ward's method uses the minimum variance criterion which minimizes the total within-cluster variance: at each step, combine two clusters whose merge results in the smallest increase in the total within-cluster variation. We determined the optimal number of clusters based on the Calinski-Harabasz (CH) index [2]. The CH index is calculated by $\frac{SS_B/(k-1)}{SS_W/(N-k)}$, where k is the number of clusters and N is the total number of cases. SS_B is the total between-cluster variance, which measures how spread apart the groups are from each other; and SS_W is the total within-cluster variance, which measures high tightly grouped the clusters are. As the number of clusters increases, SS_B keeps increasing while SS_W keeps decreasing. The CH index finds the clustering assignment that simultaneously has a large SS_B and a small SS_W by using the variance ratio criterion; the largest CH index occurs with the optimal number of clusters. The analysis was conducted using `hclust()` in R.

In order to interpret each of the different clusters, we considered the proportion vector of actions for each case. This is calculated by the frequency of the particular action (e.g., play video) divided by the count of all actions. This allows for more intuitive interpretation than using the normalized vectors and still resolves the issue that arises from varying the total number of actions for different cases. For each of the resulting clusters, we then calculated the average of the above described proportions across all the cases that were assigned to the particular cluster. Based on these metrics, we interpreted each cluster to identify distinct patterns.

4.2 Results

The resulting dendrogram from the Ward's hierarchical clustering is shown in Figure 2. The computed CH index sug-

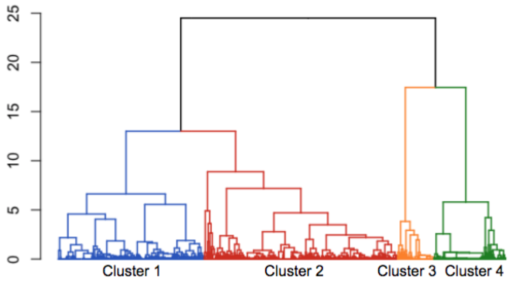


Figure 2: Dendrogram of results from the Ward's hierarchical cluster method

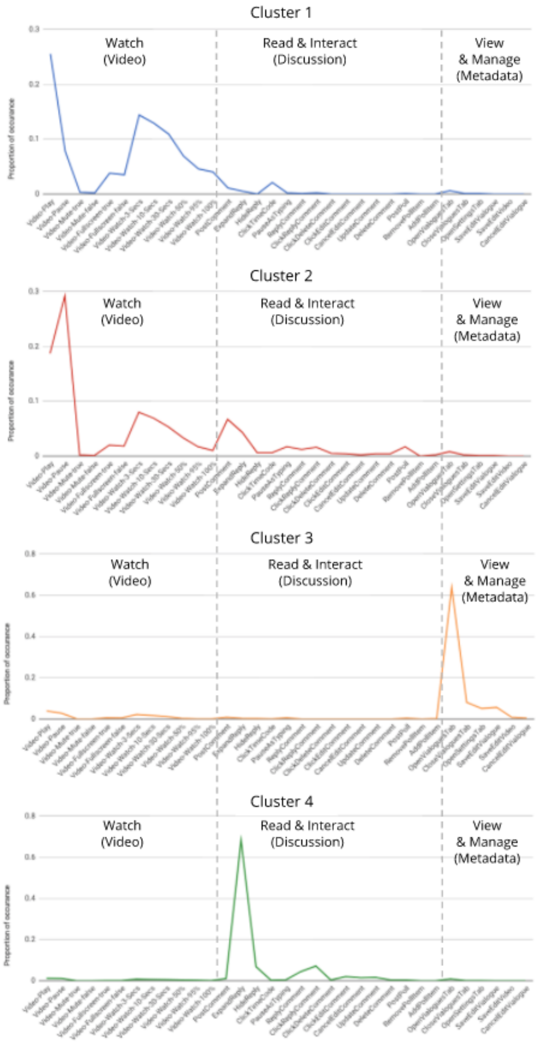


Figure 3: The four cluster profiles, or interaction patterns

gested four clusters as the best number of clusters. For each cluster, we examined the size of the cluster and the distribution of different actions based on the average proportion of each action per case, as described in a preceding method section. In order to understand the different patterns of each cluster, we plotted the average proportion of actions. Figure 3 illustrates each cluster's profile; in the x-axis, we listed different actions that exhibit similar characteristics. We first listed actions related to video watching behavior and labeled as "Watch (Video)": (in the listed order) video play, video pause, video mute true, video mute false, video full screen true, video full screen false, video watch 3 seconds, video watch 10 seconds, video watch 30 seconds, video watch 50%, video watch 95%, video watch 100%. Then, we listed those actions related to discussion activity and labeled it as "Read & Interact (Discussion)": (in the listed order) post comment, expand reply, hide reply, click time code, pause as typing, reply comment, click reply comment, click delete comment, click edit comment, cancel edit comment, update comment, delete comment, post poll, remove poll item, add poll item. As the last category, we listed actions related to viewing and creating/managing meta data and labeled as "View & Manage (Metadata)": (in the listed order) open vialogues tab, close vialogues tab, open settings tab, save edit vialogue, save edit video, cancel edit vialogue.

A graphical analysis of Figure 3 leads to the following observation: Cluster 1 shows high focus on video watching activities with no noticeable occurrence of other actions. In Cluster 2, the peaks, representing locally frequent actions, are somewhat spread out, but the graph shows the highest concentration in video watching and discussion activities. In Cluster 3, frequent actions are centered around the viewing and creating metadata. Cluster 4 shows a heavy focus on discussion activities with very limited number of other activities.

Based on the preliminary analysis of the graphs, we conducted additional examinations to understand each cluster. For the purpose of this examination, we assigned ranks based on the frequency of actions occurring in each cluster. Table 3 presents the top 10 most frequent actions on average for each cluster. In Cluster 1, the frequent actions were all related to the video watching activities: video play/pause, watch a video for a certain duration of time, and use of a full screen mode, which suggests that Cluster 1's dominating pattern is pure video watching. In Cluster 2, however, we observed that the discussion activities (post comments, expand reply) were also present in addition to video watching actions. These discussion activities were limited to the first-level interaction, and are more interactive than just watching video. However, there was limited interaction with other users/viewers. In other words, users in Cluster 2 were commenting on the video but not discussing the video with other users. Cluster 3 was unique in that the most frequent action was "Open vialogues tab" which is often used when users look for other information about the specific video such as the uploader, the upload date, and sharing features. The proportion of such actions was dominant at 0.64 while other actions' proportions were less than 0.1. Additionally, for this cluster, other actions related to editing and setting the vialogue contents were the next most frequent actions: save/edit vialogue, open settings tab, which are only allowed for content creators and moderators. Thus, the behaviors present in Cluster 3 predominantly consist of exploring the peripheral information and creating/editing vialogue metadata. In Cluster 4, the most frequent action was expanding others' replies, with the proportion of 0.69. This cluster showed the heightened focus on discussion activity in that eight of the top 10 actions were discussion-related: expand reply, click reply comment, hide reply, reply comment, click edit comment, update comment, cancel edit comment, post comment. The remaining two actions were video play/pause and no video watching for a certain period. Thus, Cluster 4 represents "opinion seeking" and "replying" behaviors.

In terms of the cluster size, the sizes were 1,137, 1,508, 282, 558 for Cluster 1, 2, 3, and 4, respectively. It is noteworthy that Cluster 2 is the largest cluster; 43% (= 1,508/3,485) of cases were assigned to Cluster 2, which was characterized as a mix of video watching and discussion activities. This behavior pattern, which combines both video and discussion, was expected to be the most popular pattern considering

Table 3: Top 10 most frequent actions and the averaged proportions in each cluster

Cluster 1 (size = 1,137)		Cluster 2 (size = 1,508)		Cluster 3 (size = 282)		Cluster 4 (size = 558)	
Video play	0.255	Video pause	0.292	Open vialogues tab	0.639	Expand reply	0.684
Video watch 3 Secs	0.144	Video play	0.187	Close vialogues tab	0.08	Click reply comment	0.071
Video watch 10 secs	0.129	Video watch 3 secs	0.08	Save edit vialogue	0.056	Hide reply	0.068
Video watch 30 secs	0.109	Video watch 10 secs	0.069	Open settings tab	0.051	Reply comment	0.044
Video pause	0.078	Post comment	0.067	Video play	0.039	Click edit comment	0.02
Video watch 50%	0.069	Video watch 30 secs	0.053	Video pause	0.028	Update comment	0.016
Video watch 95%	0.046	Expand reply	0.043	Video watch 3 secs	0.022	Cancel edit comment	0.014
Video watch 100%	0.04	Video watch 50%	0.033	Video watch 10 secs	0.017	Video play	0.012
Video full screen true	0.038	Video full screen true	0.02	Video watch 30 Secs	0.012	Video pause	0.011
Video full screen false	0.035	Video full screen false	0.018	Post comment	0.008	Post comment	0.01

that the key feature of Vialogues is its support of discussion around video content. Also noticeable was Cluster 3, which had the smallest size, with only 282 cases present (8% = 282/3,485). Cluster 3 consisted of exploring and creating metadata. Its small cluster size can be partially explained by the fact that most frequent actions of Cluster 3 were creating or editing activities available only to creators or moderators, not participants.

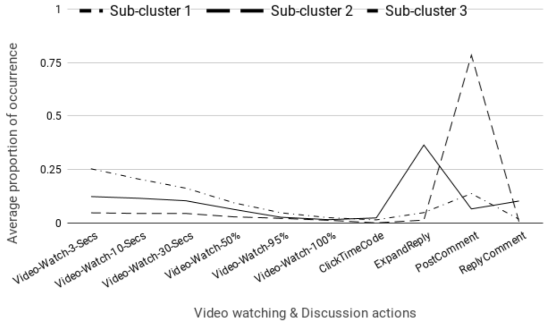


Figure 4: Profiles of three sub-clusters of Cluster 2

5. TRANSITION PATTERNS BETWEEN DIFFERENT EVENT ACTIONS

As the core objective of Vialogues is to promote discussion around video, it is important to evaluate the case in which users' usage patterns exhibit both video watching and discussion activities, e.g., identifying sequences of actions [4]. In the clustering analysis above, Cluster 2 was the largest group with both video watching and discussion, but did not show a clear classifiable pattern. Thus, we examined finer-grained user groups out of Cluster 2. In the Ward's clustering analysis, when the number of clusters increased to 6 (compared to 4 in the above analysis), Cluster 2 was further broken down into 3 clusters while Cluster 1, 3 and 4 remained as is. For the 3 sub-clusters generated from Cluster 2, using the same approach, the average proportions of actions were examined. In this case, however, we only examined the actions associated with video watching and discussion: video watch 3 secs, video watch 10 secs, video watch 30 secs, video watch 50%, video watch 95%, video watch 100%, click timecode, expand reply, post comment, and reply comment. Using only these 10 actions, the proportions of action frequency were recalculated for each action (i.e., the frequency of each action divided by the total number of frequency of 10 actions). Figure 4 presents profiles of each sub-clusters. Sub-cluster 1 represents modest amounts of both video watching and discussion, with a high proportion of posting comments among discussion activity. On the other hand, Sub-cluster 2 and 3 show heavier focus on discussion activities. Sub-cluster 2 had the highest proportion of "posting comment" action and Sub-cluster 3 had the highest peak at the "expanding replies" action.

We further examined each of the sub-clusters to understand the sequence of actions when users are involved in both video watching and discussing activities. We assumed that previous actions may potentially influence the following actions and sought to explore the path that a user takes to participate in discussion. We performed a transition network analysis, specifically applying weighted directed networks [12]. The benefit of this method is that we can discover the transition pattern between the two consecutive click activities and also gain insight about the degree by which the same action transition patterns appear. We generated weighted directed networks for each sub-cluster using the `ngram` [9] and `igraph` [3] packages in R.

The networks for three sub-clusters are presented in Figure 5. Each action name was abbreviated as follows: 's' is the indication of start, 'V-S' indicates video watching for 3, 5, 10 seconds, 'V-50' includes video watching 50%, 95%, 100%, 'PC' indicates posting comments, 'CT' is clicking timecode, 'ER' indicates expanding others' replies, and 'RC' is replying to others comments, and 'e' indicates the end of the action. The directed edges indicate transition between two consecutive actions and the width (weight) of the edges indicates the number of that transition occuring. The node size for each action indicates the total number of frequency of the particular action in the aggregated sequence set.

The left network describes transition patterns of Sub-cluster 1. Since it was characterized by a combination of a modest amount of both video watching and discussion activities, the directed edges exist for various combinations of actions with similar weights. Specifically, the transition from 's' to 'V-S' had the largest weight, indicating many users in this sub-cluster started by watching a video rather than conducting other actions. Other frequent transitions were one from 'V-S' to 'V-50' and two transitions from each of the 'V-S' and 'V-50' to 'PC'. Overall, transitions between video watching and posting comments were dominant. The network for Sub-cluster 2 is presented in the middle. For this sub-cluster, the most frequent action was posting comments. Interestingly, the graph shows heavy weights on the edge from 's' to 'PC'

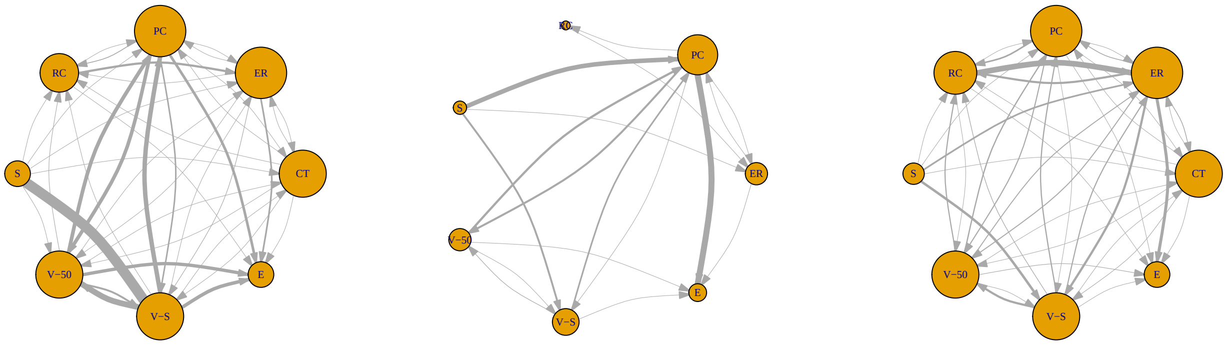


Figure 5: Weighted directed networks for Sub-cluster 1 (left), Sub-cluster 2 (middle) and Sub-cluster 3 (right)

and the one from 'PC' to 'e', which implies that users in this cluster tend to post their comment at the beginning even before watching a video and then leave the page. This could indicate that some users might read others' first level comments (which does not generate clickstream data in the current system) and then post their own comments. If this conjecture proves true, an interesting question for this group would be why are these users only posting first-level comments without replying to others' replies. This could be a future area of inquiry that helps to uncover users' path to interactive online discussion around video. Lastly, for Sub-cluster 3, the transitions from replying to comments (RC) to expanding other replies (ER) were noticeable. An interesting finding from this graph is that interactions with replies was not necessarily derived from video watching since we could not observe any significantly noticeable transition from video watching to interactive discussion with others. This suggests that for some group of users, others' comments or other factors that were not captured in clickstream data might have greater effects on replying behavior rather than the video itself.

6. CONCLUSION

In our study, we identified users' behavior patterns on Vialogues in an exploratory manner. It is important to note that while there exist a number of academic studies on the value of video based education, there are limited research papers that specifically deal with the discussion in the context of a video platform. This paper contributes to the field since it is focused on online video-based discussion, which was made possible through the Vialogues video discussion platform. Clustering analysis of different user group behaviors can provide a point of reference for future studies but more importantly, this can help educators to enhance video-based instruction and learning.

7. REFERENCES

[1] M. Agarwala, I. H. Hsiao, H. S. Chae, and G. Natriello. Vialogues: Videos and dialogues based social learning environment. In *2012 IEEE 12th International Conference on Advanced Learning Technologies*, pages 629–633, July 2012.

[2] T. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics – Theory and Methods*, 3(1):1–27, 1974.

[3] G. Csardi and T. Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[4] D. Ferreira, M. Zacarias, M. Malheiros, and P. Ferreira. Approaching process mining with sequence clustering: Experiments and findings. In G. Alonso, P. Dadam, and M. Rosemann, editors, *Business Process Management*, pages 360–374, 2007.

[5] K. J. Lee and M. D. Sharma. Incorporating active learning with videos: A case study from physics. *Teaching Science: The Journal of the Australian Science Teachers Association*, 54(4), 2008.

[6] F. Murtagh and P. Legendre. Ward's hierarchical agglomerative clustering method: Which algorithms implement ward's criterion? *Journal of Classification*, 31(3):274–295, 2014.

[7] D. Passey. *Digital video technologies enhancing learning for pupils at risk and those who are hard to reach.*, pages 156–168. Glasgow Caledonian University Press, 2006.

[8] O. Poquet, L. Lim, N. Mirriahi, and S. Dawson. Video and learning: A systematic review (2007–2017). In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, LAK '18, pages 151–160, New York, NY, USA, 2018. ACM.

[9] D. Schmidt and C. Heckendorf. *Guide to the ngram Package: Fast n-gram Tokenization*, 2017. R Vignette.

[10] S. Schwan and R. Riempp. The cognitive benefits of interactive videos: learning to tie nautical knots. *Learning and Instruction*, 14(3):293–305, 2004.

[11] I. Vieira, A. P. Lopes, and F. Soares. The potential benefits of using videos in higher education. In *Proceedings of EDULEARN14 Conference*, pages 0750–0756. IATED Publications, 2014.

[12] S. Wasserman and K. Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

Is the Doer Effect Robust Across Multiple Data Sets?

Kenneth R. Koedinger
Human-Computer Interaction Institute
Carnegie Mellon University
koedinger@cmu.edu

Richard Scheines
Department of Philosophy
Carnegie Mellon University
scheines@cmu.edu

Peter Schaldenbrand
Human-Computer Interaction Institute
Carnegie Mellon University
pschalde@cs.cmu.edu

ABSTRACT

The "doer effect" is the assertion that the amount of interactive practice activity a student engages in is much more predictive of learning than the amount of passive reading or watching video the same student engages in. Although the evidence for a doer effect is now substantial [6, 7, 12], the evidence for a causal doer effect is not as well developed. To address this, we mined data for evidence of a causal doer effect across multiple domains. We examined data from two online courses in Psychology, one in Biology, one in Statistics, and two in Information Science, applying causal discovery algorithms [14] in Tetrad to each. Assuming that factors driving a student's choices regarding how to spend their time in an online course are temporally prior to their performance on quizzes and exams, we found evidence of a causal relationship in every domain we studied. We did not find evidence that a unique causal model held in every domain we studied, but when we estimated the size of the causal relationships in the models we found in each domain, we did find evidence in every case that doing has a much stronger quantitative effect on learning than either reading or watching video. This work may be the first EDM effort to explore the generalizability of a causal claim about learning across multiple datasets from a variety of courses and contexts of use. It makes vivid the role of causal data mining algorithms in educational research. The evidence presented furthers the case for doer effect causality, but also recommends a need for richer data with more student background and learning process variables to better isolate causal directionality without assumptions about temporal order and unmeasured confounds.

Keywords

Doer effect; learning by doing; causal discovery

1. INTRODUCTION

When students take an online course, or use a cognitive tutor, a log of data is created that records their interactions with the course or tutor. Mining this data for causal information concerning what sorts of student behaviors cause better learning outcomes is crucial if we are to intervene, either on the design of the online material, or on the student's behavior more directly.

In this paper, we explore the causes of learning in several online courses using Tetrad and Tigris/LearnSphere. Tetrad (http://www.phil.cmu.edu/tetrad/) is a causal discovery tool that has already proved helpful in educational data mining [6, 10], and LearnSphere is a collaboration dedicated to providing data and tools for analyzing information pertaining to student learning (http://learnsphere.org/). LearnSphere combines data and analysis tools with Tigris, a workflow tool that connects data from the educational data repository DataShop [5] to analytical programs such as Tetrad. Tigris runs in a web browser and has functionality to use the abilities of Tetrad and share results of analyses with other Tigris users. Tigris allows users to test theories across diverse datasets, and this was precisely our goal in the work we describe here. Tigris connects analytical tools to data and users via their research. LearnSphere users can upload datasets to DataShop [5] and make them available in workflows. They can also share their own analytics as well as workflows they construct in Tigris. The causal models and analysis in this paper were executed using the Tetrad implementation in Tigris.

The causal discovery algorithms in Tetrad operate on graphical causal models [14], which allow us to rigorously represent the qualitative causal structure of a domain with a directed graph, and to connect the structure of the graph to statistical constraints that we can test on measured data. The algorithms compute the equivalence class of causal structures that are consistent with background knowledge about the domain. In some cases the equivalence class is not very informative - for example the equivalence class of a system of two variables X,Y that are correlated is: $X \rightarrow Y$, $X \leftarrow Y$, $X \leftarrow \text{Confounder} \rightarrow Y$. In systems involving more than 2 variables, the causal information from an equivalence class can be much more informative.

The question of how to judge whether or not to believe an equivalence class output by the algorithms is very complicated and very interesting. All models within an equivalence class have the same "fit" with data, but whether the statistical fit is "good enough" to warrant belief depends on a large number of factors. This is by no means a problem that is special to causal discovery algorithms, however, and it is not the subject of our work. It is one that should concern all data-mining procedures, including ones that involve a single human building a hypothesis and then testing it on a single dataset.

Our concern in this paper is whether or not evidence for a causal doer effect generalizes across courses and contexts. We studied courses with diverse subject matter and diverse student populations.

The "doer effect" is the assertion that the amount of interactive practice activity a student engages in is much more predictive of learning than the amount of passive reading or watching video the same student engages in. We want evidence of a causal doer effect, that is, intervening to increase the amount of interactive practice would result in better learning outcomes.

Previous work has provided some evidence for a causal doer effect. In [12], 52 students at the University of Pittsburgh took an online course in which five variables were measured: pretest, percent of modules printed, percent of interactive exercises completed as a measure of "doing", average end of module quiz score, and score on final exam.

Printing out modules was convenient and more common among good students, but it reduced the likelihood that students would complete interactive exercises (they could not do these on the printed modules). It thus served as an "instrument" for the doing → Quiz → Final exam relationship.

This relationship between performing active assignments and a learning outcome was directly researched in [6] and coined the "doer effect" in [7]. A dataset with six variables was examined in [6]. In this data, the relationship between doing and learning was far stronger than the relationship between passive activities such as watching videos or reading course material and learning.

Furthering the evidence for the doer effect, in [7], the relationship was tested on four other datasets, using regression methods. These were a diverse set of courses, but all had shown a strong link between doing and learning. While a strong correlation between doing and performance was shown in [7], the causal relationship was not tested. In this paper, we extend the investigation of whether the doer effect is causal by explicitly employing causal discovery techniques in Tetrad to these additional datasets.

We examined relationships between approximately six variables that are persistent throughout course subject matter, student populations, and time. Our research question: Is there evidence that the doer effect is causal across multiple contexts/datasets?

2. RELATED WORK

Much of the EDM research has investigated correlational relationships in predictive models. In [11], correlations of variables predict whether a student will enroll in college. While having a successful predictor of college attendance is good, it would be more useful to educators to understand the causes of college attendance so they can make interventions and increase applications and yield. In [13], correlation mining is used to explore a relationship between the features of a math problem and student learning. They acknowledge that future work would have to go into determining if these relationships are causal. Only once the relationships are determined to be causal can they assuredly be used to influence course design. Analyzing whether these relationships are causal by performing a randomized assignment experiment is the gold standard for making causal inferences, but this is often impractical, and there are thousands of non-experimental datasets available with which we can test the external validity (or generalizability) of hypotheses across multiple contexts [8]. Thus, it is worthwhile to pursue the use of causal discovery methods designed for non-experimental data on such datasets [6, 9].

Research into students' attitudes toward a math tutor [4] conclude that correlations exist between empathetic messages in the tutor and a student's mood toward it. They suggest that the positive correlation they found is indicative of a relationship in which increasing the empathy of these messages would cause a better mood amongst the users of the tutor. This implies a causal relationship, but they do not consider confounding variables or causal discovery algorithms [14].

Previous work in EDM that has researched causal relationships include [3] and [9]. Both of these use causal discovery algorithms and [9] uses Tetrad. Rather than resource use variables found in this paper, [3] uses variables that measure a student's interest and actions in a tutor, and it provides evidence for causal relationships between these variables and a final exam grade.

These past efforts [3, 6, 9, 12] have performed analyses on single datasets and, as such, there remains an opportunity to use the vast number of datasets available to probe external validity. This paper is distinctive in this regard -- to our knowledge, this is the first EDM effort to explore the generalizability of a causal claim about learning across multiple datasets from a variety of courses and contexts of use.

3. METHODS: CONFIRMATORY & EXPLORATORY WITH CRITERIA

We pursued both confirmatory and exploratory approaches to addressing our research question by analogy, for example, to confirmatory and exploratory factor analysis [15].

3.1 Method 1: Confirmatory Analysis of Causal Model Generality

Our confirmatory analysis involved testing a causal model that displayed the doer effect that was derived from data aggregated from a class offered at Georgia Tech in 2013, (https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=863) on five other datasets. We tested if the model statistically fits each dataset, according to the goodness-of-fit measures common in linear causal models [14]. We know of no successful attempts to test a specific causal model discovered on one dataset on other datasets collected in widely varying contexts, as in our datasets which have different kinds of course activities collected in different educational settings and with different available measures of student performance and different sizes of data. In attempting this confirmatory analysis, we discovered that it was neither going to confirm nor deny the doer effect hypothesis. We present it nevertheless as a cautionary message for others who may be tempted to do the same and to explain how dataset variations, particularly dataset size, make inferences from a confirmatory analysis problematic.

The causal model in Figure 1a was the model discovered on data from a 2013 Georgia Tech psychology course [6]. The model was previously [6] discovered using the Tetrad Java application, but in this paper, the analysis was performed using Tetrad's implementation in LearnSphere's Tigris workflow tool resulting in the same model structure, with negligible edge coefficient differences. The dataset features six variables measured on 939 students. One variable is a prior knowledge assessment (Pretest), one is a measure of doing in terms of the number interactive activities students performed (activities_started), two are measures of student use of passive learning resources including text page reading (non_activities_pageview) and video watching (play), and two are measures of learning outcome including the total across 11 unit quizzes (T.Quiz) and a final exam score (Fina_Exam). A directed edge in a causal model depicts evidence of a direct causal relationship between the variables. The coefficient on the edge is an indication of the strength of the causal relationship.

The primary feature to note in the causal model in Figure 1a is that while the outcome measures (T.Quiz and, indirectly, Fina_Exam) are effects both of passive resource use (non_activities_pageview and play) and active resource use (activities_started), it is the active resource use that exhibits the much stronger relationship. This large difference (0.44 vs. .06) is the doer effect. It is also important to note that the edges in the model do not represent correlations between the variables; they express and quantify direct causal relationships. For example, while activities_started and Fina_Exam have a correlation coefficient of 0.28, the causal inference algorithm determines they do not have a direct causal relationship.

Table 1. How the various naming schemes of datasets relate to each other.

	Psychology Georgia Tech	UMUC: Bio, Psych, Stat, InfoSci	C@CM
Pre-assessment	Pretest		Pretest
Doing activities	activities_started	activities_started	activities_started
Reading text pages	non_activities_pageview	non_activities_reading	non_activities_pageview
Watching lecture videos	play		
Unit level assessments	T.Quiz	total_quiz_proportion	T.Quiz
Cumulative assessment	Fina_Exam	final_grade_in_number	C@CM_Final_Exam

It does so by finding that when conditioned on T.Quiz, Fina_Exam and activities_started are independent.

A final note is to emphasize that the causal claims are about the *constructs* being measured not about the *measures* themselves. For example, the causal link between T.Quiz and Fina_Exam indicates that better competence attained during the course (the construct that T.Quiz measures) causes better competence at the end of the course (the construct that Fina_Exam measures). It *does not* imply that merely raising a T.Quiz (e.g., by making the quiz easier) would cause final exam scores to increase

A difficulty with testing a model on different datasets is the fluctuating naming schemes of variables and the inconsistency with which variables are contained within datasets. For instance, GTech's psychology dataset contains seven variables while a dataset from The University of Maryland University College, which is also used in this paper, has four variables. The four variables in the UMUC data are a subset of GTech's psychology data. For each dataset, we used the closest set of variables we could construct. Table 1 shows our decisions.

To facilitate comparison across datasets in the confirmatory analysis, we used the maximum number of variables that were common to the original dataset and the dataset being tested. We used five variables when we tested the original model on C@CM and four variables when we tested it on the UMUC datasets.

While we received UMUC data from the previous study [7], we added a sixth dataset from an online course on basic computing offered at Carnegie Mellon which we call Computing@Carnegie Mellon. A pre-assessment variable was created for each student by averaging the highest scored attempt at each pre-assessment quiz. The same process was performed on unit level assessments for each student. The number of active activities was the number of activities that each student started, and the number of passive activities was calculated in the same way as [6]. For a student to get to an activities page, they needed to visit a readable page. To accurately represent the number of pages read by a student, the total number of readable pages each student visited was subtracted by the number of activities they performed divided by a ratio. This ratio was the number of activities started to the total pageviews of the student with largest number of activities started. Therefore, the page viewing variable would not quantify the pages that students viewed merely as a stepping stone to get to activities. Once we made these datasets compatible with GTech's data, we could test our original model on five datasets.

3.2 Method 2: Exploratory Analysis with Criteria

Our second pass at answering our research question involved exploratory analysis whereby we applied a causal discovery algorithm to each dataset instead of confirming the original model on the other datasets. In this approach, we don't expect to find the same model on each dataset, but we do hope to see evidence of a causal doer effect in each context. We asked the question: What are the properties of the search output that would constitute evidence of the causal doer effect? These properties will be the criteria that we use to determine if each different context provides evidence of a causal doer effect. We identified them as:

Properties of a causal model exhibiting evidence of the causal doer effect.

1. There exists a causal edge between doing and either of the outcome measures that has a positive coefficient estimate.
2. The strength of this causal edge is larger than all the edges from passive resource use to the outcome measures.
3. The edge(s) between doing and outcome(s) is oriented *from* doing *to* an outcome.

4. RESULTS

We now provide results from the two methods, first the confirmatory analysis and then the exploratory analysis.

4.1 Confirmatory Analysis: Testing a Causal Model Across Multiple Datasets

In order to determine if the causal model discovered on GTech's psychology course data would fit other datasets, modifications to the data were made to ensure that all datasets were comparable. We show in Figure 1 the causal model that was used as a "modified original" causal model, which was in turn then tested on new data. We arrived at the "modified original" model by applying the same causal search algorithm to the original data set – but with the set of variables that were common to both it and the dataset to be tested. Happily, these models are strongly consistent with the original. For instance, when the play variable was removed, the value of the edge from non_activities_pageview to T.Quiz (i.e., non_activities_pageview→T.Quiz) should be adjusted. This adjustment should be equal to the original edge between these variables plus the product of the edges from the two edges that were removed (i.e., non-activites_pageview→play and play→T.Quiz).

Table 2. The causal model that was discovered on GTech's psychology dataset was estimated using data from datasets listed in the first row of the table.

	UMUC Biology	UMUC Info Sci	UMUC Psychology	UMUC Statistics	C@CM	UMUC Biology (sample)	UMUC Info Sci (sample)
#Students	3516	6112	89	61	383	300	300
Chi-square	78.89	18.44	1.04	28.33	14.30	11.92*	3.32*
DOF	2	2	2	2	4	2	2
P-value	0	0	0.59	0	0	0.02*	0.49*

*average of multiple trials with different samples

non_activities_pageview→T.Quiz + (non_activities_pageview→play * play→T.Quiz) = edge's new value

0.0650 + (0.1149*0.0645) = 0.0724

The model estimated the new value for the edge from non_activities_pageview to T.Quiz to be 0.0713, which is consistent with the calculation above.

The causal models in Figure 1 show yellow edges. These edges were originally either unoriented in the representation of the equivalence class or were oriented as bidirected edges. Before we can estimate and test a causal model, we must direct all edges to form a directed acyclic graph. Therefore, before estimating, the undirected and bidirected edges were *arbitrarily* converted into directed edges - and such edges are shown in yellow to caution the user against inferring any directional information from such edges. In Figure 1c, if the edge were directed the opposite way, the coefficient would still be 0.3927. Removing variables such as play and Pretest still allowed for models that show strong doer effect to be discovered, which is consistent with [6].

The structures of the models in Figure 1 were then applied to the other five datasets, and these models were estimated to determine how well the exact causal structure of the "original model" fit the new data. The results of the confirmatory analysis are summarized in Table 2. As was expected, whether the original causal structure fit other datasets was inconsistent. UMUC's psychology dataset fit very well to this causal model having a p-value of 0.59, however, the rest of the p-values from full datasets were low. It is worth noting that the only full data set to fit GTech's psychology course, was another psychology course. UMUC and GTech's psychology courses have the same content (online readings and interactive activities). The differences between these datasets were the population that created the data and the number of variables. GTech's course had all of the variables that UMUC's course had with the addition of the number of videos watched and a pretest. Therefore, once the video watching and pretest variables are removed from GTech's psychology dataset, the same causal model would be expected to be discovered on GTech's and UMUC's psychology data.

In large datasets, e.g., with N > 2000, the goodness-of-fit χ^2 statistic is of limited use, as it not only tests for causal structure, but it also becomes sensitive to small deviations from linearity, or normality, or other parametric assumptions that have little to do with the causal structure. To test whether the statistic is rejecting the model structure or fine-grained violations of the parametric assumptions, we took a random sample of 300 students from each of the UMUC datasets and then re-estimated and tested the model. The smaller biology sample showed a much better fit than its full dataset, however, at a p-value of 0.02, the model is still rejected. The sample from the information science course showed an excellent fit with a p-value of 0.49 and a chi-square value that differs from the

Figure 1. Using subsets of variables from the Georgia Tech Psychology dataset, three causal models were discovered using the PC algorithm and an alpha value of 0.05 as in [6].

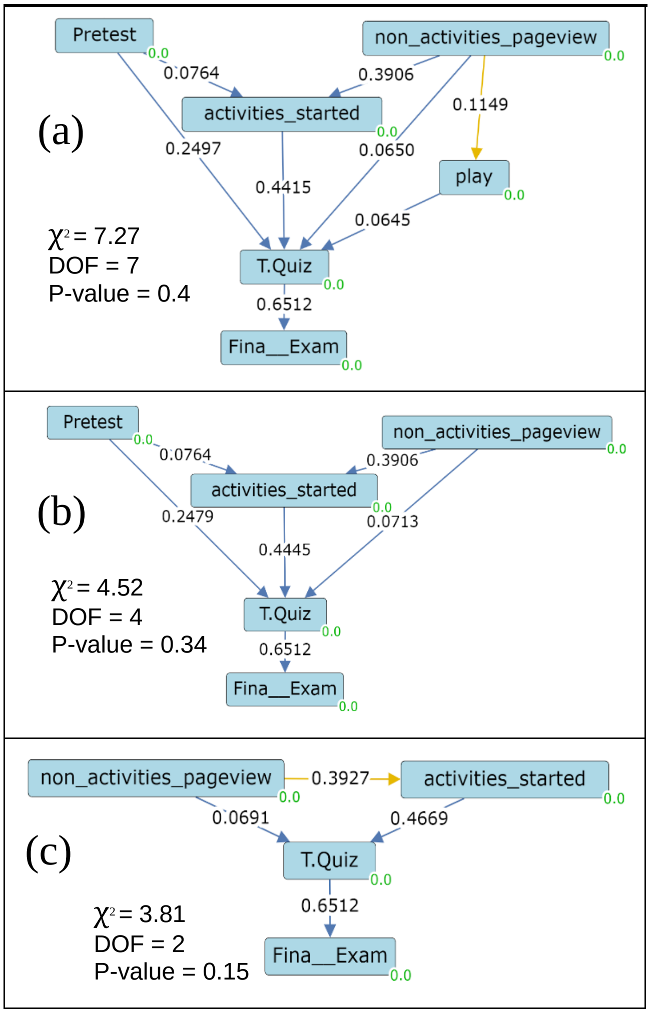
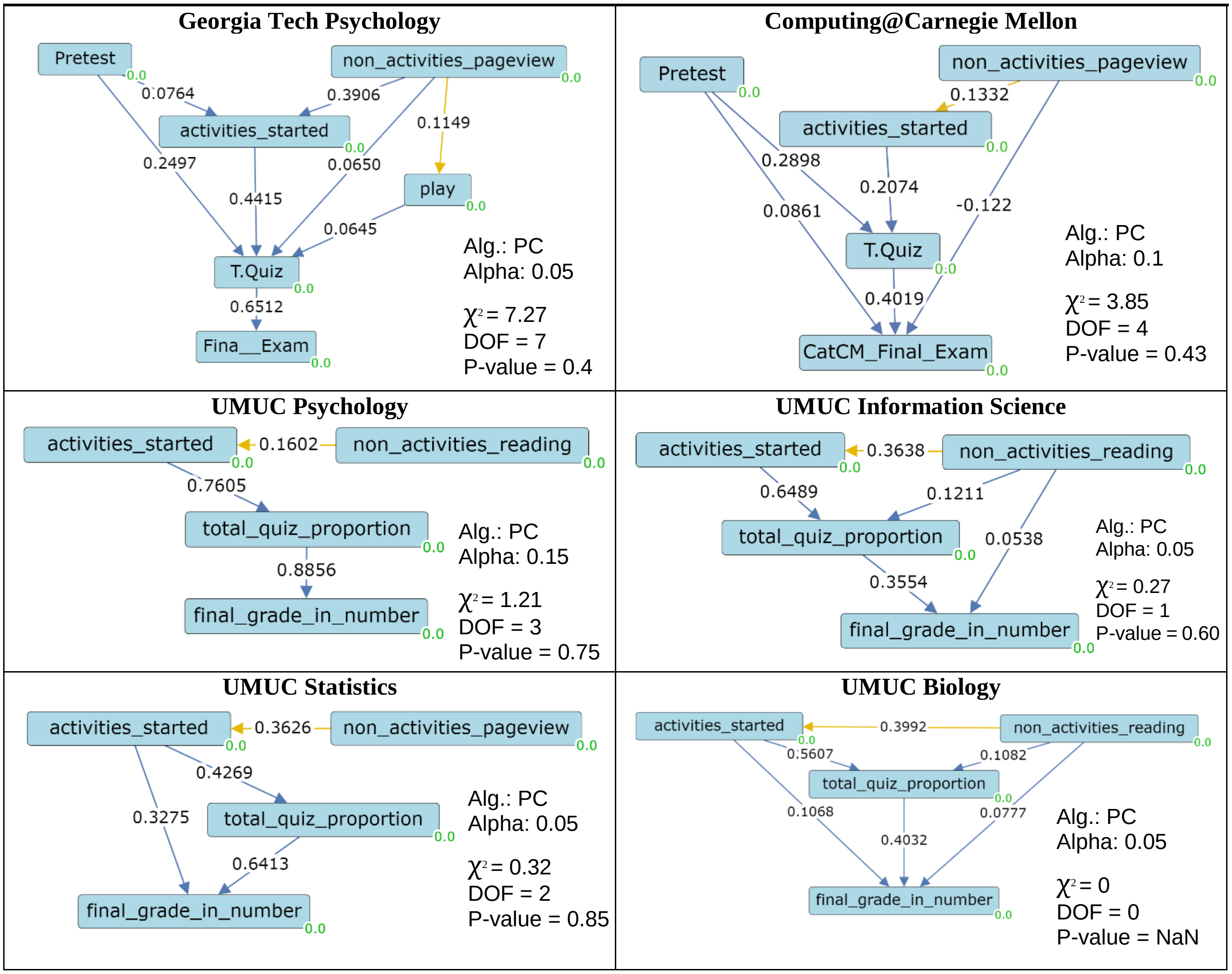


Figure 2. Causal models of various datasets. To the bottom right of each model are the search algorithm and p-value cutoff for searching (alpha) used to discover the model. Below that are the model statistics when estimating the model on the dataset: Chi-square (χ^2), degrees of freedom in the model (DOF), and p-value.



degrees of freedom of the model by only 1.32. We take this to be evidence, although only weak evidence, that the causal structure in the "original model" is reasonably consistent with the measured data. This marginal fit exceeds expectations given the history of difficulty in fitting single models across domains. Given the diversity of the datasets and lack of control between them, any indication of generalizability adds to external validity even though the fit was marginal.

4.2 Exploratory Analysis: Causal Doer Effect Criteria Across Multiple Datasets

The inconsistencies in fitting a single, exact causal model across such diverse datasets are to be expected. A more targeted approach focuses the evaluation on just the variables of interest for assessing the causal doer effect. As described above, we defined three criteria to indicate whether a model provides causal evidence for the doer effect. We searched for causal models on each dataset and then evaluated them by these criteria. Unlike the confirmatory strategy (as shown in Figure 1), where models were discovered on one dataset and estimated on another, these models were discovered and estimated on the same data, as is the norm in causal discovery and as was done previously [6, 12].

Figure 2 shows the results of this analysis. For every dataset we discovered a model that fit the data well (with the exception of Biology, where the model discovered is untestable because it entails no constraints and thus has 0 degrees of freedom). The causal model discovered in [6] was found using the PC algorithm with a p-value cutoff (alpha) of 0.05 for detection of reliable links between variables. This is the algorithm and alpha value that produced a model with largest p-value upon estimation – indicating the model does not significantly deviate from the data and thus is a good one. For the datasets in Figure 2, we also used the PC algorithm with alpha = .05, .1, or .15.

In order to assess the goodness-of-fit of the whole model, we use the p-value of the χ^2 statistic [1]. Unlike the usual logic in hypothesis testing, the p-value in this context uses a null of the specified model. So, a low p-value indicates that we should reject the specified model, while a p-value over .05 indicates that we cannot reject the specified model from the data measured. In general, the χ^2 test is more tolerant of simple models, and simple models are also favorable since they only show the strong, important edges.

The models in Figure 2 were discovered using the same many-tiered prior knowledge as the models in Figure 1 and Table 2. This prior knowledge assumes that the pre-assessments and weekly/unit assessments were taken before and after the doing and passive activities, respectively. This is an assumption that prohibits causal directionality that violate the temporal order, but it is *not* an assumption that a causal edge exists. That is, the assumption does not guarantee that the algorithms will find any edge between doing and learning. If it does find an edge, then it will be directed from doing to outcome as opposed to vice versa.

Setting these tiers for input in the search algorithms in Tetrad dictates that if a causal link is to be found between variables between temporal tiers, then the directionality of the edge will be from the tier earlier in time to the tier later in time. Again, putting the doing variable in an earlier tier than an outcome variable does not guarantee that Tetrad will find a causal link between the two variables.

We then asked whether the models discovered from each data set satisfy any of the three properties that indicate a causal doer effect as we had listed before. Analyzing Figure 2, all six datasets we used in this paper produced causal models that meet all three criteria of a model with a causal doer effect. For example, C@CM's causal model has a directed edge with a coefficient of 0.2074 from doing to an outcome measure, therefore displaying the first and third properties. The coefficient from the only other resource use variable (non_activities_pageview) was -0.122. The strength of the causal edge is larger than the edge from passive resource use to the outcome in C@CM, thereby showing the second property. The model for UMUC's biology class is not testable as a model, as it has 0 degrees of freedom. Nevertheless, the model along with the estimated coefficients on the edges support all three criteria of a causal doer effect.

5. DISCUSSION

We build off of the work in [6] by providing evidence to suggest that the doer effect is indeed causal. Data from a variety of different online courses (Psychology, Computing, Information Science, Statistics, and Biology) and course use scenarios (MOOCs and for-credit college courses), analyzed with causal discovery algorithms all provide evidence that the doer effect is causal and not just associational.

The correlation between doing and outcome is interesting, but establishing the correlation does not specify whether an intervention on doing would affect outcome. If the doer effect is causal, then modifying learning environments to guide or encourage students to spend more time engaging in interactive activities will result in more learning.

In addition to finding evidence for a causal relationship between doing and learning, we articulated what we hope are useful new methods for discovering and testing for cause-effect relationships across diverse datasets.

For our confirmatory strategy, we tested models discovered in one context on data from another. Finding models that fit a held-out subset of data is protection against overfitting – but it does not mean that those models will fit datasets collected in entirely new contexts, in fact, it is nearly impossible to fit across datasets as diverse as these. Although models developed for educational research seem unlikely to fit in new contexts, we found that features of the causal model of the doer effect found in Georgia Tech data did seem to generalize. The specific model discovered on Georgia Tech's Psychology course data fit extremely well on the data from UMUC's Psychology data. The courses had the same content, but they had different students and were offered in quite different settings (MOOC vs. for-credit course). A marginal fit of the causal model from GTech's Psychology course onto UMUC's Biology and Information Science courses provides some support, albeit limited, for even broader generalization of a specific causal model across different contexts. Given that task has been shown to be nearly impossible, these results are significant even though most fits were marginal.

The inconsistencies of fitting a specific model across contexts is not an indication that a causal doer effect is not present throughout the contexts, it is, however, an indication that an exact model is inconsistently present throughout the contexts. The difficulty of fitting a specific model across contexts led us to reconsider this confirmatory approach. Although a fully specified causal model failed to generalize, it appeared to be due to differences in links between variables that are not relevant to the main question of whether the doer effect is causal. Thus, we developed a method to examine just the key claims of the target theory, in our case, a theory of a causal doer effect. We did so by generating a causal model in an exploratory fashion for each dataset and then evaluating the resulting model as to whether it fit the key criteria for providing evidence of the doer effect.

In all datasets we found that: 1) there was a positive causal edge between active doing and either of the outcome measures, 2) the strength of this causal edge was larger than all edges from passive resource use (reading and watching) to the outcome measures, and 3) the edge(s) between active doing and outcome(s) was oriented from doing to an outcome.

This work provides many possible subsequent inquiries. One area of future work is to test the assumption on the directionality of the causal link between doing and learning outcome. In this paper, we used temporal knowledge to constrain the search algorithms to direct a causal relationship, if one was found between doing and outcome, to be directed from doing to outcome. This temporal knowledge does not make it more likely to find that there is an edge between doing and outcome, it only constrains its orientation. The fact that we found a causal edge between doing and outcome in all six domains is exciting, but we need to investigate further to see if the direction of these edges can be determined from the data or from other plausible assumptions.

When we relax the assumption that doing is temporally prior to outcome, Tetrad is not as likely to orient the edges between doing and learning. Unlike the dataset from Pitt described in the introduction [12], where we were lucky to find a natural "instrument," we do not have a variable in the datasets we studied that is likely to take on that role. Identifying a broader set of variables in this dataset (e.g., by distinguishing counts of error-free doing from errorful doing) or in other datasets may lead such a natural instrument. Particularly useful datasets would involve more student background variables, such as demographics and prior aptitudes, as well as more detailed process data, such as when scrolling makes parts of a web page, whether text, video, or activity, visible or not to a student.

We also hope to perform an experiment to test and hopefully confirm the causal doer effect, much as Rau, et al., [10] did by performing an experiment to test hypotheses generated with causal discovery algorithms on non-experimental data.

6. ACKNOWLEDGMENTS

This work was supported by a National Science Foundation grant (ACI-1443068).

7. REFERENCES

[1] K. Bollen, Structural Equations with Latent Variables, John Wiley & Sons, 1989.

[2] D. Chickering, Optimal Structure Identification with Greedy Search, in *Journal of Machine Learning Research* 3 (2002) 507-554

[3] S. Fancsali, Causal Discovery with Models: Behavior, Affect, and Learning in Cognitive Tutor Algebra, *in Proc. of the 7th International Conf. on Educational Data Mining*, 2014.

[4] S. Karumbaiah, B. Woolf, R. Lizarralde, I. Arroyo, D. Allessio and N. Wixon, Addressing Student Behavior and Affect with Empathy and Growth Mindset, in *Proc. of the 10th International Conf. on Educational Data Mining*, 2017.

[5] K. Koedinger, R. 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,* Boca Raton, CRC Press, 2010.

[6] K. Koedinger, J. Kim, J. Jia, E. McLaughlin and N. Bier, Learning is not a spectator sport: Doing is better than watching for learning from a MOOC, in *ACM Conf. Learn at Scale*, 2015.

[7] K. Koedinger, E. McLaughlin, J. Jia and N. Bier, Is the Doer Effect a Causal Relationship? How Can We Tell and Why It's Important, in *Conf. Learning Analytics and Knowledge*, 2016.

[8] J. Pearl and E. Bareinboim, External Validity: From Do-Calculus to Transportability Across Populations, *Statistical Science*, vol. 29, no. 4, pp. 579-595, 2014.

[9] D. Rai, J. Beck and I. Arroyo, Causal Modeling to Understand the Relationship between Student Attitudes, Affect and Outcomes, in *Proc. of the 6th International Conf. on Educational Data Mining*, 2013.

[10] M. Rau, R. Scheines, V. Aleven and N. Rummel, Does Representational Understanding Enhance Fluency – or Vice Versa? Searching for Mediation Models, in *Proc. of the 6th International Conf. on Educational Data Mining*, 2013.

[11] M. San Pedro, R. Baker, A. Bowers and N. Heffernan, Predicting College Enrollment from Student Interaction with an Intelligent Tutoring System in Middle School, *in Proc. of the 6th International Conf. on Educational Data Mining*, 2013.

[12] R. Scheines, G. Leinhardt, J. Smith and K. Cho, Replacing Lecture with Web-Based Course Materials, *Journal of Educational Computing Research*, vol. 32, no. 1, pp. 1-26, 2005.

[13] S. Slater, R. Baker, J. Ocumpaugh, P. Inventado, P. Scupelli and N. Heffernan, Semantic Features of Math Problems: Relationships to Student Learning and Engagement, *in Proc. of the 9th International Conf. on Educational Data Mining*, 2016.

[14] P. Spirtes, C. N. Glymour, R. Scheines, Causation, Prediction, and Search, 2nd edition, MIT Press 2000.

[15] B. Thompson, Exploratory and confirmatory factor analysis: Understanding concepts and applications. Washington, DC, US: American Psychological Association. 2014 http://dx.doi.org/10.1037/10694-000.

Understanding Learners’ Opinion about Participation Certificates in Online Courses using Topic Modeling

Gaurav Nanda[a] Nathan M. Hicks[a] David R. Waller[a] Dan Goldwasser[b] Kerrie A. Douglas[a]
[a] School of Engineering Education, [b] Department of Computer Science, Purdue University, West Lafayette, USA

ABSTRACT

This study proposes a formal multi-step methodology for qualitative assessment of topic modeling results in the context of online learner motivation to purchase Statements of Participation (SoP). We developed Latent Dirichlet Allocation (LDA) based topic models on open-ended responses of three post-course survey questions from 280 open courses offered on the FutureLearn learning platform. For qualitative assessment, we first determined the theme of the topic based on the words that constituted the topic and responses that were most strongly associated with the topic. Then, we verified the theme by comparing the topics assigned by LDA model on a test set with manual annotation. We also performed sentiment analysis to check for alignment with human judgment. Learner motivations in each theme were interpreted with the Expectancy-Value-Cost framework. Our analyses indicated that, primarily, learners were motivated to purchase the SoP based on perceptions of the utility value and financial cost of the certificate. We found that human judgment agreed with the topic model more frequently when LDA topic weights were larger.

Keywords

MOOC Certificates, Topic Modeling, Latent Dirichlet Allocation, Text Mining

1. INTRODUCTION

Open-ended survey responses contain rich information that is often hard to capture through closed-ended questions. Open-ended questions allow users to not only answer the question asked but also express their opinions freely, offer insights that may be novel, and provide suggestions for improvement. For an evolving system such as Massive Open Online Courses (MOOCs), where there is a large variation in the learners’ backgrounds and learning objectives, it is challenging to design closed-ended surveys with predetermined options encompassing all aspects. Therefore, use of open-ended surveys that allow obtaining detailed feedback and insights from users on different aspects can be very useful. However, manually analyzing open-ended survey responses from large, diverse populations can be challenging. Data mining techniques can be helpful in this regard, but they involve issues related to interpretability of their results.

In the context of our research, the primary issue is the extent to which topics identified by topic modeling techniques represent qualitatively meaningful themes.

1.1 Topic Models

While manual analysis of open-ended responses is extremely tedious, topic modeling algorithms can find emerging themes from a large collection of documents [1] and have been used for exploratory analysis of large textual collections such as MOOC discussion forums [2]. In this study, we used Latent Dirichlet Allocation (LDA) based topic modeling, which is a probabilistic unsupervised classification method that models each document as a mixture of underlying topics and each topic as a collection of related words. The LDA model tries to identify these topics iteratively based on the co-occurrence of words in documents and represents each document as a composition of different topics with associated weights. A good explanation of the algorithm can be found in [3]. “Topic models provide useful descriptive statistics for a collection, which facilitates tasks like browsing, searching, and assessing document similarity” [4].

Notably, the topic model algorithms have no domain knowledge and the documents are not annotated with topics or keywords. However, the generated topics often resemble the thematic structure of the document collection and topic annotations by model are useful for tasks such as classification and data exploration. “In this way, topic modeling provides an algorithmic solution to managing, organizing, and annotating large archives of texts” [5].

Since topic modeling is an unsupervised method, the ground truth set of topics is unknown—which makes it hard to judge the quality and relevance of topics identified by models such as LDA. Also, the interpretability of the topics generated from these models is not guaranteed [6]. Measures such as Perplexity or Probability of held-out documents [7] have been proposed for evaluating the quality of topic models but they have not been found to correlate well with human judgment because they do not capture topic coherence or semantic interpretability [8], [9]. On the other hand, ‘Topic Coherence’ measures have been found to better correlate with human judgment [6], [10], [11]. Finding out the exact meanings of the topics requires additional information and domain knowledge [12]. In a study comparing human evaluation of topics with these traditional metrics, authors recommended that “practitioners developing topic models should thus focus on evaluations that depend on real-world task performance, rather than optimizing likelihood-based measures” [8]. Therefore, in this study, we conducted qualitative analysis of topics identified by LDA model to determine their theme and relevance in context of online course certificates.

1.2 Online Participation Certificates

MOOCs provide the opportunity to deliver knowledge and skills to learners anywhere in the world, at relatively low cost. Learners can document their MOOC achievements through certificates, which are increasingly becoming an acceptable medium for skill or knowledge validation among employers [13], [14]. It has also been found that learners who opt for certification in MOOCs are more likely to actively participate in and complete courses [14],

[15]. As such, identifying factors associated with certificate purchasing can lead to better participation and learning.

Our aim in this study was to understand the value that learners associate with the course participation certificate. To our knowledge, previous literature has not studied large-scale learner feedback to assess the importance of online learning certificates. In this study, we analyzed the open-ended responses to post-course survey questions from about 280 courses offered on the FutureLearn platform to understand the reasons why learners were interested or not interested in the Statement of Participation (SoP), and what would make it more appealing to them. On the platform used for this study, the SoP can be purchased by learners if they "mark over 50% of the steps on a course as complete and attempt all test questions" [16].

1.3 Learner Motivation and the Expectancy-Value-Cost Model

The Expectancy-Value-Cost (EVC) model of motivation has been shown to capture the important features of learning, persistence, and performance-based behaviors. EVC theory characterizes motivation to engage in a given task by the expectation of success, the perceived value, and the perceived cost of engaging in the task [17]. Expectancy is related to a learner's self-conception of their ability, task difficulty, and academic mindset, and it helps predict achievement. Value is based on intrinsic motivation, perceived utility, and attainment (affirmation of identity), and it is highly related to continued interest and persistence. Cost has four associated elements related to task effort, outside effort, loss of valued alternatives (including money), and emotion. Cost negatively affects both expectancy and value in different ways [18]. Because retention in MOOCs is a common problem, this study aims to understand the values and costs associated with SoPs which can help increase MOOC completion rates.

When learners decide to participate in MOOCs, they come with a wide variety of backgrounds and motivations. Their varying circumstances affect their ability to invest time, effort, and money to participate, and through EVC theory these variations can help develop our understanding and strategies to increase motivation, such as offering the chance to invest in a SoP [19], [20]. However, there are a variety of influences on learners' decisions to purchase SoPs. When a learner enrolls in a MOOC and purchases the SoP, their investment is often associated with its value and cost and can provide a motivational tool for learning and course completion. Thus, the reasons why learners do or do not purchase SoPs can inform this motivational strategy for improved retention and learning.

2. METHOD

We analyzed following three post-course survey questions:

Q1. Why are you interested in a SoP?
Q2. If no (not interested in SoP), why not?
Q3. What would make a SoP more appealing to you?

The post-course survey data was provided to us by the platform in the form of separate CSV files for each course. We first collated together all the responses to each of the listed questions from different courses. From the collected responses, we removed the records that did not contain any text. It is to be noted that considerably more learners answered the post-course survey question Q2- why they were not interested in the SoP (~56,000), than Q1-why they were interested in it (~12,600). It was encouraging that a lot of learners (~49,000) answered Q3-what would make the SoP more appealing to them. Regarding the length of responses, about 30% of responses for Q1 and Q2, and 40% for Q3, had 5 or fewer words. For all questions, about 60% responses had 10 or fewer words and about 75% responses had 15 or fewer words. For each question, we randomly selected 100 responses to be used as the TEST set and the remainder to be the TRAIN set.

2.1 Topic Modeling

We used the MALLET library [21] for developing the LDA topic models for each question using the respective TRAINING set. During the model development, stopwords that were in the MALLET Stopword list were removed. We did not perform stemming of words and considered only single words. The LDA model requires the number of topics to be provided as an input. We conducted a preliminary analysis by providing 10 topics as input and qualitatively examining the words that constituted the topic and responses that were strongly associated with each topic. We observed that some of the topics were very similar which indicated that the optimal number of topics was fewer than 10. To determine the optimal number of topics, we used the CV_Coherence measure using the package PyLDAvis [22], as earlier studies have found CV_Coherence to be well-correlated with human judgment. We compared the CV_Coherence values of different number of topics between 5 and 10 and selected the optimal number of topics as the one with highest CV_Coherence for each question. Subsequently, LDA models were developed on the TRAINING dataset for all three questions. MALLET provides following outputs that were used for qualitative analysis:

a) A list of the top words that constitute each topic. For example, for topic T_i, the list of the top k words, $W_i = \{w_i^1, w_i^2, ..., w_i^k\}$, that constitute the topic are outputted. The value of k was set to be 20 for this study.
b) The composition of each document (open-ended responses, in our case) in terms of topics and associated weights. For example, for given topic model with n topics $\{T_1, T_2, ..., T_n\}$, the composition of a response R_i is represented as: $C(R_i)= p_i^1T_1 + p_i^2T_2 + p_i^3T_3+ ... + p_i^nT_n$, where p_i^j represents the relative weight associated with topic T_j and the sum of all topic weights for a document is one. Therefore, documents composed of multiple topics are expected to get assigned smaller weights for multiple topics, and documents composed of a single topic are expected to have a high weight associated for that topic.

2.2 Qualitative Analysis

The objective of qualitative analysis of the topics generated by the LDA model was to validate the understanding of underlying themes. The qualitative analysis involved the following steps:

1) First, two researchers developed initial themes for each topic from the list of top words that constituted the topic. Then, the 100 responses with the largest weights for that topic were examined to check if they corresponded to the initial theme and the themes were updated if any missing aspects were discovered. Thus, the themes were iteratively developed by sampling more instances. We selected high weight examples for theme development as they were composed mainly of a single topic of interest. To illustrate this process, one of the topics that emerged from the responses to Q3 (What would make SoP more appealing to you?) comprised the following words: *free, cheaper, cost, price, charge, expensive, print, download, version, pay, lower, certificate, online, bit, downloadable, digital, purchase, statement, pdf, copy.* By inspecting the words in context of the question asked, we can

deduce that this topic was related to the SoP cost being too expensive and a downloadable, digital copy would be a good alternative. Then, by examining strongly associated responses with this topic, such as, *"A more affordable price point. Possibly this could be done by having the option of a downloadable certificate so would save on printing, packaging, and postage,"* we could confirm that the theme we developed for the topic was appropriate but should include that a digital certificate would be considered a cheaper option.

2) The next step was to evaluate the LDA model trained on the TRAIN dataset by assessing its topic-assignment on the TEST dataset, which was not used to train the LDA model. The responses in the TEST set were manually annotated with up to three most likely topics, then checked if the top topic assigned by the LDA model was among those three. Notably, it was difficult to manually assign only one topic to responses in the TEST set, as many topics contained overlapping ideas. This is discussed further in the Results section. We also studied the relationship between the weight associated with the top topic and the level of agreement between the LDA model and human judgment.

3) We also performed qualitative analyses of responses that were composed of multiple topics according to the LDA model to further test our understanding of the topic theme. For each question, we randomly selected 100 sample cases where the LDA output had two topics with weights greater than 0.4. A researcher, who was blinded to the topic-composition assigned by the model, annotated the cases with two most prominent topics. The manual annotation was compared with the topic-composition of LDA model.

4) Sentiment analysis was performed on the responses and the sentiment-polarity of the responses associated with each topic was examined as an additional validation. We used the Natural Language Toolkit NLTK Vader sentiment intensity analyzer [23], [24], that is pre-trained on a large corpus of annotated social media text and outputs a score for Positive, Negative, and Neutral sentiments. The average sentiment score for each topic was determined by averaging the Positive, Negative and Neutral sentiment scores of responses with that as top-topic. Next, we examined whether the sentiment scores were consistent with the expected prevalent sentiment of the topic or not.

3. RESULTS

We observed the highest CV_Coherence at 6 topics for Q1 and Q2, and at 5 topics for Q3. Therefore, these were selected as the optimum number of topics and provided as input to the LDA topic model. The topics that emerged for Q1, Q2 and Q3, their themes and top-10 words, are presented in Table 1. The qualitative and sentiment analyses of topics for each question are discussed below.

3.1 Q1: Why are you interested in Statement of Participation?

The LDA model identified six topics describing interest in the SoP. Table 2 summarizes the results of qualitative and sentiment analyses for Q1. The column "%Top Topic" indicates the percentage of cases in the TRAIN and TEST datasets where that topic was the top topic. The column "%Agree-TRAIN" indicates the percentage of cases among the top 100 cases of that topic in the TRAIN dataset where the response was consistent with the theme of the topic. The column "%Agree-TEST" indicates the percentage of cases for each topic where the top topic assigned by the LDA topic model was among the three topics assigned manually. The column "Average Sentiment Score-TRAIN" indicates the average score of Positive (Pos in Table 2), Negative (Neg in Table 2) and Neutral (Neu in Table 2) sentiments as outputted by the NLTK Sentiment Intensity Analyzer for all the responses in the TRAIN dataset that had the respective topic as the top topic identified by the LDA model.

Table 2. Qualitative and Sentiment Analyses Summary: Q1

Topic	**%Top Topic**		**%Agree**		**Average Sentiment Score-TRAIN**		
	Train	Test	Train	Test	Pos	Neg	Neu
Q1T1	29	25	87	80	0.11	0.01	0.88
Q1T2	33	38	84	71	0.14	0.01	0.85
Q1T3	16	0	96	0	0.07	0.00	0.92
Q1T4	13	38	86	42	0.16	0.01	0.83
Q1T5	6	0	59	0	0.17	0.02	0.81
Q1T6	4	0	77	0	0.14	0.01	0.84

The agreement of the theme of the topics with human judgment in the TRAIN set was relatively good (close to 90%) for all the topics except topic Q1T5. However, we did not observe a similar level of agreement between the topic predicted by the topic model and manual annotation in the TEST set. One of the primary reasons for this effect is that the 100 responses reviewed manually in the TRAIN set had a considerably high topic-weight (>0.85) while the weights of top-topic in the TEST set were not as high (being as low as 0.28 for some cases). For the qualitative analysis of 100 responses that were mostly composed of two topics, we found that a) for 18% of the cases, the model and human judgment agreed for both topics, b) for 64% of the cases, only one of the topics assigned by the model and human agreed, and c) for the remaining 19%, neither of the two topics assigned by the model and human agreed.

Given the positive framing of Q1, the expected prevalent sentiment in learners' responses was positive or neutral, but not negative. The sentiment analysis also agrees with expectations. The responses within each topic were predominantly classified as neutral (81-92%) and positive (7-17%). It is to be noted that the NLTK sentiment analyzer, trained on annotated media corpus differing from our dataset, may produce somewhat noisy results.

Based on topic themes for Q1 as shown in Table 1, it seems that learners would be interested in obtaining the SoP if they perceive a) personal attainment value and/or a high time or effort cost for the course, for example, keeping the SoP as a memento of their hard work, b) professional utility value, such as demonstrating interest in an area to employers and universities, or c) low financial cost of the SoP and high utility or interest value of the courses, wanted to contribute back to the platform for providing great learning experiences free of charge.

3.2 Q2: If not interested in Statement of Participation, why not?

The LDA model identified six topics related to learners' disinterest in the SoP, as described in Table 1.

Table 1: List of Topics, their Themes and Top-10 Words for Q1, Q2 and Q3

Topic	Theme of the topic	Top 10 words
Q1T1	Learners wanted SoP as a proof of completing the course for personal (record of their personal achievement of finishing the course) and professional (a good addition to their resume) reasons.	record, participation, achievement, proof, cpd, completed, personal, part, add, work
Q1T2	Learners want to demonstrate their interest in a particular area for professional purposes, such as applying to universities for higher studies or demonstrating interest or skills to a potential future employer.	future, show, career, interest, proof, job, knowledge, study, university, work
Q1T3	For many learners who were working professionals, the SoP fulfilled their work-related requirement of "continuous professional development (CPD)" or training hours.	development, professional, cpd, evidence, learning, portfolio, continuing, personal, work, education
Q1T4	They wanted SoP as a reminder of the great learning experience or the time and effort they put in the course. They perceived interest or attainment value in the SoP and recognized a high time or effort cost for the course. They also wanted to show it to family and friends with pride.	time, show, learning, work, put, learn, reminder, effort, good, I've
Q1T5	Given that the courses are offered for free on the platform, learners who could easily afford to pay for the SoP wanted to support the platform so that it could continue to offer courses for free.	courses, certificate, free, pay, FutureLearn, statement, money, it's, feel, back.
Q1T6	Learners felt the SoP would be professionally useful due to various reasons, such as the course being related to their area work or coming from a reputable university.	history, university, interested, knowledge, health, teaching, work, college, education, science
Q2T1	Learners did the course out of personal interest in the subject or for leisure. They were either retired or the course was not related to their professional field.	interest, retired, personal, don't, certificate, career, participation, learning, prove, feel.
Q2T2	The price of the SoP seemed expensive to learners and they could not afford it at that time due to their financial situation.	money, purchase, buy, afford, expensive, moment, time, courses, future, cost.
Q2T3	Some learners did not need the SoP as a) they already had advanced degrees, b) they were very experienced professionally, or c) they were retired.	paper, retired, don't, certificates, knowledge, certificate, learn, learning, piece, interested.
Q2T4	Learners were not sure about the worth of SoP as it a) indicated only participation in the course and did not specify course accomplishments, learning, scores, or level of engagement, or b) was not clearly recognized by employers and universities.	certificate, participation, statement, completed, feel, complete, didnt, time, purchase, work
Q2T5	The current price of the SoP seemed high to learners due to different reasons such as their financial situation, or high currency exchange rates (if they lived in developing countries).	expensive, free, certificate, pay, cost, bit, paper, price, high, courses
Q2T6	It was difficult for international learners to buy the SoP due to high currency exchange rates and non-availability of convenient payment methods. Learners mentioned that payment through credit card or international bank transfer was not easy in their country.	card, credit, pay, payment, country, money, don't, online, bank, live.
Q3T1	Learners suggested that a) SoP should be cheaper, b) digital version of SoP should be downloadable for free, and payment should be needed for a formally verified hard copy, d) pricing should be based on the country, e) more payment methods such as *PayPal* should be supported, and f) there should be option to choose soft copy or hard copy of SoP as shipping may be difficult and costly for remote locations	free, cheaper, cost, price, charge, expensive, print, download, version, pay.
Q3T2	SoP would be more appealing if it were more relevant for their career or job, such as being recognized by employers as qualification or counting as CPD.	career, work, needed, don't, statement, job, interest, participation, retired, relevant.
Q3T3	This topic had two themes: a) the price of the SoP was too high for which some learners suggested membership model and subsidized costs for low income learners; and b) learners were not sure how to answer Q3 as some had got the SoP and some didn't want it as they did the course for recreation	courses, free, don't, appealing, money, cost, make, statement, paper, answer
Q3T4	Learners suggested that instead of showing just participation, the SoP should show detailed course achievements to properly reflect their efforts and achievements	statement, participation, certificate, completed, test, level, completion, achievement, score, tests
Q3T5	Learners would be interested in buying the SoP if it was more recognized professionally, such as course credits, recognition by employers and valid continuous professional development. Some learners suggested a more formal look of SoP with university logo.	university, qualification, recognized, credit, credits, certificate, courses, points, degree, academic

Similar to Table 2, Table 3 presents the distribution of topics, level of agreement between the topic assignment by LDA model and manual annotation, and the average sentiment scores for each topic for Q2. As shown in Table 3, the level of agreement with the human annotation in the TRAIN set is not consistently higher than the TEST set, and for some topics, it is higher for the TEST set.

Table 3. Qualitative and Sentiment Analyses Summary: Q2

Topic	**%Top Topic**		**%Agree**		**Average Sentiment Score-TRAIN**		
	Train	Test	Train	Test	Pos	Neg	Neu
Q2T1	38	58	100	79	0.11	0.06	0.83
Q2T2	25	15	80	71	0.05	0.06	0.89
Q2T3	15	12	72	100	0.08	0.07	0.85
Q2T4	12	5	65	67	0.07	0.07	0.86
Q2T5	8	6	65	100	0.08	0.06	0.86
Q2T6	3	4	93	75	0.06	0.10	0.84

Some of the possible reasons for this behavior, which is considerably different from Q1 (as shown in Table 2), may be: a) the higher number of responses for Q2 (55,000) as compared to Q1 (12,600), which may lead to samples in the TEST set being more similar to TRAIN set, and b) greater level of overlap between the topics generated for Q2 as compared to Q1. To illustrate the latter point, as shown in Table 1, there seems to be considerable amount of overlap between the themes of topics Q2T2, Q2T5, and Q2T6, with all being related to the financial cost of the SoP. This may cause the LDA model to assign either of these topics as top-topic based on the words present in the response. Additionally, these topics are highly likely to be assigned as top-3 topics during manual annotation of responses in the TEST involving cost aspect of the SoP. Therefore, it is likely to result in a higher level of agreement between manual annotation and top-topic assigned by LDA model in TEST set.

For the qualitative analysis of 100 responses that were mostly composed of two topics, we found that a) for 30% of the cases, the model and human judgment agreed for both topics, b) for 56% of the cases, only one of the topics assigned by the model and human agreed, and c) for the remaining 14%, neither of the two topics assigned by the model and human agreed. We observed higher level of agreement for top-two topics as compared to Q1.

Given the negative framing of Q2, the prevalent sentiment of responses was expected to be between neutral and negative. The sentiment scores for Q2 in Table 3 indicate that the responses were largely neutral in nature. We did not observe relatively higher score for Negative sentiment as compared to Positive sentiment (in fact, for some topics such as Q2T5, Positive had a higher average score). This differed from our expectation about the prevalent sentiment in Q2 responses.

Based on topic themes for Q2 as shown in Table 1, it seemed that learners would not opt for SoP if they perceived a) high financial or effort costs, or b) low utility or attainment value, as they did the course for leisure or did not benefit from it professionally.

3.3 Q3: What would make a SOP more appealing to you?

For Q3, the five topics that emerged from the LDA topic model are presented in Table 1. As expected, Q3 topics were similar in theme to Q2 topics, as, in Q3, learners suggested approaches to address the concerns they mentioned in Q2. Similar to Tables 2 and 3, Table 4 summarizes the distribution of topics, agreement between LDA model and manual annotation, and the average sentiment scores for Q3.

Table 4. Qualitative and Sentiment Analyses Summary: Q3

Topic	**%Top Topic**		**%Agree**		**Average Sentiment Score-TRAIN**		
	Train	Test	Train	Test	Pos	Neg	Neu
Q3T1	35	46	90	65	0.16	0.06	0.77
Q3T2	26	16	82	94	0.11	0.05	0.84
Q3T3	17	12	80	83	0.12	0.06	0.82
Q3T4	14	13	80	54	0.10	0.04	0.85
Q3T5	9	13	83	85	0.11	0.02	0.86

As shown in Table 4, we observe a high level of agreement between the LDA model and human judgment for most topics in the TRAIN set, and for all other topics except Q3T1 and Q3T4 in the TEST set. For Q3T1, the lower level of agreement in the TEST set may be due to considerable overlap in the themes of Q3T1 and Q3T3 on the cost aspect of SoP. Similarly, there is overlap in themes of topics Q3T4, Q3T5, and Q3T2 regarding the professional recognition of the SoP by employers.

For the qualitative analysis of 100 responses that were mostly composed of two topics, we found that a) for 49% of the cases, the model and human judgment agreed for both topics, b) for 44% of the cases, only one of the topics assigned by the model and human agreed, and c) for the remaining 6%, neither of the two topics assigned by the model and human agreed. We observed a higher level of agreement for top-two topics in Q3 as compared to Q1 and Q2.

Our expectation of the prevalent sentiment of Q3 responses was between neutral and positive and not as negative as Q2. The sentiment scores for Q3 responses are similar to Q1, with relatively high score for Neutral, followed by Positive, and then Negative. In summary, the learners suggested that they would be more inclined to buy the SoP if it were more affordable, recognized professionally, detailed their accomplishments and learnings; and convenient payment options were available.

4. DISCUSSION

In this study, we analyzed large number of open-ended responses using LDA topic model followed by qualitative analysis of the topics to determine and verify the topic-themes. It is important to mention the limitations associated with our study. While the topic model brought up some prominent themes from the responses, there may be other important themes that did not get highlighted because of low frequency. Therefore, the results from the topic model are not exhaustive and cannot replace detailed manual qualitative analysis that can identify such themes. It is to be noted that the topic themes were not distinct in nature and had overlapping elements with other topics, for example, in Q2, there were multiple topics on the financial cost of the SoP. During manual review process, we also noticed that learner responses often involve multiple topics and the weights assigned by the LDA model for prevalent topics may not represent the actual composition strength of the topic.

We also observed a consistent pattern for all questions that the top-topic predicted by LDA model in the TEST dataset agreed better with human annotation when the weight of the top-topic (as assigned by the LDA model) was higher. This is represented in Figure 2 as the plot between the weight of top-topic (shown as w) with agreement between LDA model and human annotation for TEST datasets of Q1, Q2, and Q3.

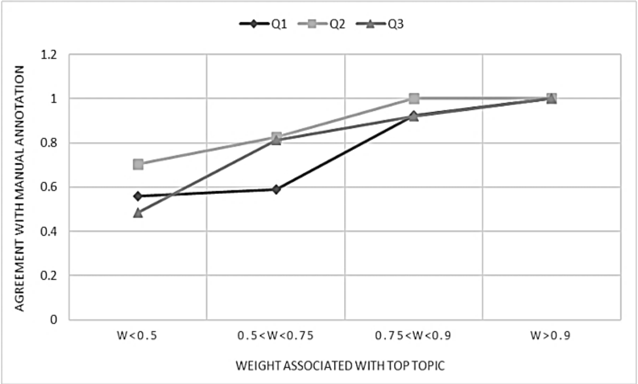


Figure 1. Weight of top-topic and level of agreement with human annotation in TEST dataset for Q1, Q2, and Q3

As shown in Figure 1, there is a relatively low level of agreement between the topic model and human judgment when the top-topic weight is less than 0.5, but picks up in the range of 0.5-0.75, and is extremely high when the weight is more than 0.75.

From the topic model analysis, there were some clear connections with aspects of value and cost in EVC theory. As expected, the expectancy dimension of motivation was not relevant for these questions. For learners for who purchased the SoP, interest, utility, and attainment values were associated with personal and career related considerations and the reputation of those offering the MOOCs, while costs were associated with task effort and time commitment. Complimentary to these findings, reasons for not purchasing the SoP were the perceived lack of value for both current and future needs, but cost focused, primarily, on the financial expense, even when high values were expressed. The suggestions for making the SoP more appealing also centered around motivational aspects of increased value and professional utility and decreasing financial or effort costs.

5. IMPLICATIONS

The implications of this study relate to the methodology of qualitative validation of topic models and learner motivations to purchase SoPs.

5.1 Methodology

Manual analysis of open-ended responses involves multiple steps such as developing a coding scheme and then coding the data, which can be challenging for large numbers of responses. Topic models provide an effective means for exploratory data analysis for a large collection of textual data but mostly require qualitative analysis for interpretability. Our results indicated that the proposed methodology for qualitative evaluation of topics generated by LDA is reliable and can be replicated for similar studies involving large-scale open-ended survey data. We also found that the topics predicted by the LDA model were more likely to agree with human judgment if the weight assigned by the LDA model was higher (>0.75). This indicates that the weight assigned by the LDA model is in line with human judgment. Still, the probabilistic nature of the LDA algorithm is such that the weights may not be perfectly representative of the composition of themes present in a response, particularly when topics are highly overlapping or consist of disparate sub-themes.

5.2 Learner Motivation

Given there is a large variation in background and learning objectives of online learners, their need for certification also varies. Research indicates that participants who pay for certification have a higher completion rate than students who choose to audit the course. Furthermore, the majority of participants report that they intend to fully participate in all aspects of the course; however, most do not fulfill this commitment. Therefore, it is important to understand what learners feel about participation certificates to improve the offering by platforms and to take advantage of the motivational benefits of certificates to increase course completion.

Based on the topics generated from learner responses, we obtained the following insights about learners' opinions of course participation certificates: a) learners were interested in buying the SoP if they valued it personally or professionally or wanted to contribute to the platform, b) learners were not interested in buying the SoP if they thought it was too expensive, lacked utility value, or were taking the course for purely recreational reasons, and c) learners believed the SoP would be more appealing if it were professionally recognized, adequately reflected effort, and cost less.

6. CONCLUSIONS

Our results showed that our multi-step approach for qualitative analysis is robust as there was high level of agreement between human judgment and topic assignment by the LDA model when the model assigned larger weight to the topic–which meant that the theme developed for the topic in the first step of qualitative analysis was appropriate. This approach for qualitative analysis of topic models would be applicable for similar studies analyzing large amounts of textual data.

This study examined how learners perceive the value of online learning certificates based on their responses to post-survey questions. It is worth mentioning that the post-course survey was taken only by learners who completed the course and not all enrollees. Future work may involve collecting feedback from all enrollees about certification in online courses that may lead to insights on their motivations for the course.

We found that one group of learners reported value in obtaining the certificate and appreciated the artifact to keep of their learning. However, another group of learners cited cost and lack of value as main reasons for not opting in for the certificate. One potential explanation may be the individual learner's socio-economic status or country location and their ability to pay for the MOOC.

MOOCs were founded as affordable learning opportunities; however, many learners indicated the certificate was priced out of their range. While obtaining a certificate may increase a learner's participation in a course and provide documentation of their achievement, it must be priced at an amount that learners world-wide can afford.

EVC theory provided a useful interpretive lens for the motivational aspects of investing in a SoP, which can be used to inform strategies for encouraging this investment and increasing course completion. Future studies could examine employer perceptions of MOOC certificates and ways of increasing the credibility of learning in a MOOC.

7. ACKNOWLEDGEMENTS

Our thanks to FutureLearn for providing us the post-course survey data. This work was made possible by the National Science Foundation (NSF) (PRIME #1544259). 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.

8. REFERENCES

[1] C. Wang and D. M. Blei, “Collaborative topic modeling for recommending scientific articles,” presented at the *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’11*, New York, New York, USA, 2011, p. 448. doi: 10.1145/2020408.2020480.

[2] A. Ramesh, D. Goldwasser, B. Huang, H. Daume, and L. Getoor, “Understanding MOOC Discussion Forums using Seeded LDA,” presented at the *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications*, Stroudsburg, PA, USA, 2014, pp. 28–33. doi: 10.3115/v1/W14-1804.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet Allocation,” *J. Mach. Learn. Res.*, vol. 3, no. Jan, pp. 993–1022, 2003. [Online]. Available: http://www.jmlr.org/papers/v3/blei03a.html.

[4] J. D. Mcauliffe and D. M. Blei, “Supervised Topic Models,” presented at the *Advances in Neural Information Processing Systems (NIPS 2007)*, 2007, pp. 121–128.

[5] D. M. Blei and D. M., “Probabilistic topic models,” *Commun. ACM*, vol. 55, no. 4, p. 77, Apr. 2012. doi: 10.1145/2133806.2133826.

[6] M. Röder, A. Both, and A. Hinneburg, “Exploring the Space of Topic Coherence Measures,” presented at the *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining - WSDM ’15*, New York, New York, USA, 2015, pp. 399–408. doi: 10.1145/2684822.2685324.

[7] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno, “Evaluation methods for topic models,” presented at the *Proceedings of the 26th Annual International Conference on Machine Learning - ICML ’09*, New York, New York, USA, 2009, pp. 1–8. doi: 10.1145/1553374.1553515.

[8] J. Chang, S. Gerrish, C. Wang, J. L. Boyd-graber, and D. M. Blei, “Reading Tea Leaves: How Humans Interpret Topic Models,” presented at the *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, 2009, pp. 288–296.

[9] D. O ’Callaghan, D. Greene, J. Carthy, and P. Cunningham, “An analysis of the coherence of descriptors in topic modeling,” *Expert Syst. Appl.*, vol. 42, no. 13, pp. 5645–5657, 2015. doi: 10.1016/j.eswa.2015.02.055.

[10] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin, “Automatic evaluation of topic coherence,” presented at the *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Los Angeles, California, 2010, pp. 100–108.

[11] D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum, “Optimizing semantic coherence in topic models,” presented at the *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Edinburgh, United Kingdom, 2011, pp. 262–272.

[12] H. Bisgin, Z. Liu, H. Fang, X. Xu, and W. Tong, “Mining FDA drug labels using an unsupervised learning technique - topic modeling,” *BMC Bioinformatics*, vol. 12, no. Suppl 10, p. S11, 2011. doi: 10.1186/1471-2105-12-S10-S11.

[13] H. C. Davis, K. Dickens, M. Leon Urrutia, M. del M. Sanchéz Vera, and S. White, “MOOCs for Universities and LearnersAn analysis of motivating factors,” presented at the *6th International Conference on Computer Supported Education*, 2014.

[14] J. Friedman, “5 Reasons to Consider Paying for a MOOC Verified Certificate | Online Colleges | US News,” 2016. [Online]. Available: https://www.usnews.com/education/online-education/articles/2016-03-04/5-reasons-to-consider-paying-for-a-mooc-verified-certificate. [Accessed: 02-Oct-2017].

[15] T. R. Liyanagunawardena, P. Parslow, and S. Williams, “Dropout: MOOC participants’perspective,” presented at the *EMOOCs 2014, the Second MOOC European Stakeholders Summit*, Lausanne, Switzerland, 2014, pp. 95–100.

[16] “FAQ- FutureLearn.” [Online]. Available: https://about.futurelearn.com/about/faq?category=certificates-and-statements. [Accessed: 28-Sep-2017].

[17] K. E. Barron and C. S. Hulleman, “Expectancy-Value-Cost Model of Motivation,” in *International Encyclopedia of the Social & Behavioral Sciences*, Elsevier, 2015, pp. 503–509.

[18] J. K. Flake, K. E. Barron, C. Hulleman, B. D. McCoach, and M. E. Welsh, “Measuring cost: The forgotten component of expectancy-value theory,” *Contemp. Educ. Psychol.*, vol. 41, pp. 232–244, Apr. 2015. doi: 10.1016/J.CEDPSYCH.2015.03.002.

[19] T. D. Reeves, A. A. Tawfik, F. Msilu, and I. Şimşek, “What’s in It for Me? Incentives, Learning, and Completion in Massive Open Online Courses,” *J. Res. Technol. Educ.*, vol. 49, no. 3–4, pp. 245–259, Oct. 2017. doi: 10.1080/15391523.2017.1358680.

[20] D. Koller, A. Ng, and Z. Chen, “Retention and Intention in Massive Open Online Courses: In Depth,” *Educause Review*, pp. 62–63, Jun-2013.

[21] A. K. McCallum, “MALLET: A Machine Learning for Language Toolkit.” [Online]. Available: http://mallet.cs.umass.edu/. [Accessed: 29-Sep-2017].

[22] “Welcome to pyLDAvis’s documentation! — pyLDAvis 2.1.1 documentation,” 2015. [Online]. Available: http://pyldavis.readthedocs.io/en/latest/. [Accessed: 21-Dec-2017].

[23] “NLTK Sentiment Analysis.” .

[24] C. J. Hutto and E. Gilbert, “VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text,” presented at the *Eighth International AAAI Conference on Weblogs and Social Media*, 2014.

Predicting Student Enrollment Based on Student and College Characteristics

Ahmad Slim
University of New Mexico
Albuquerque, NM 87131, USA
ahslim@unm.edu

Don Hush
University of New Mexico
Albuquerque, NM 87131, USA
dhush@unm.edu

Tushar Ojah
University of New Mexico
Albuquerque, NM 87131, USA
tushar3309@unm.edu

Terry Babbitt[*]
University of New Mexico
Albuquerque, NM 87131, USA
tbabbitt@unm.edu

ABSTRACT

Colleges are increasingly interested in identifying the factors that maximize their enrollment. These factors allow enrollment management administrators to identify the applicants who have higher tendency to enroll at their institutions and accordingly to better allocate their money rewards (i.e., scholarship and financial aid). In this paper we identify factors that affect the likelihood of enrolling. We use machine learning methods to statistically analyze the enrollment predictability of such factors. In particular, we use logistic regression (LR), support vector machines (SVMs) and semi-supervised probability methods. The LR and the SVMs methods predict the enrollment of applicants at an individual level whereas the semi-supervised probability method does that at a cohort level. We validate our methods using real data for applicants admitted to the university of New Mexico (UNM). The results show that a small set of factors related to student and college characteristics are highly correlated to the applicant decision of enrollment. This outcome is supported by the relatively high prediction accuracy of the proposed methods.

Keywords

Student enrollment, student characteristics, college characteristics, classification, logistic regression, support vector machines, time series analysis, variable selection

1. INTRODUCTION

In the past years enrollment management emerged as an important structure in academic institutions [3]. Its direct influence on the performance of such institutions made it a cornerstone. Don Hossler, John P. Bean, and colleagues defined enrollment management as ”an organizational concept and a systematic set of activities designed to enable educational institutions to exert more influence over their student enrollments. Organized by strategic planning and supported by institutional research, enrollment management activities concern student college choice, transition to college, student attrition and retention, and student outcomes. These processes are studied to guide institutional practices in the areas of new student recruitment and financial aid, student support services, curriculum development, and other academic areas that affect enrollments, student persistence, and student outcomes from college” [5].

A direct consequence of this process is the major involvement of enrollment management in budgeting and financial aid planning. This requires that administrators of the enrollment management communicate with administrators of the financial aid office to better allocate scholarship and financial aid rewards in order to maximize enrollment. Considering the large expenditure on the scholarship and the financial aid awards, this research explores different factors that presumably influence the enrollment decision of applicants. The intention of this work is to provide decision makers in the enrollment management administration a better understanding of the factors that are highly correlated to the enrollment process. These factors might better identify the applicants who have higher tendency to enroll at an institution relative to others. This allows enrollment management to assign money rewards efficiently and thus not only maximize enrollment but also save a big portion of institutional money. These factors basically include a wide range of features related to student characteristics and institutional characteristics.

For this purpose, we use real data for applicants admitted to the university of New Mexico (UNM) as a case study. UNM represents a variety of regional comprehensive universities and thus the results of this work could be widely applicable to other universities. UNM is a public research university in Albuquerque, New Mexico. It is the largest post-secondary institution in the state in total enrollment across all campuses and one of the state's largest employers. The acceptance rate at UNM is 45% with an average enrollment of 3,500 new beginning student per year [2].

[*] Associate Vice President of Enrollment Management

The results in this work are presented using machine learning methods and data mining techniques. We use logistic regression (LR) and support vector machines (SVMs) models in addition to time series analysis and probability approaches.

The remainder of this paper is organized as follows. Section 2 presents a descriptive definition of nationally effective student and institutional characteristics in addition to others that are introduced for the first time in the literature. Section 3 introduces our proposed models. Section 4 shows some experimental results. Finally, Section 5 presents some concluding remarks.

2. FEATURE DESCRIPTION

The data set used in this work contains more than fifty features for admitted students at UNM. These features describe some of the student and the college characteristics. These features are represented by a set of binary, categorical, discrete and continuous variables. The section below lists a description for each of these features and explains the intuition guiding us to include them in our analysis.

- GENDER: a binary variable indicating the sex of the applicant (i.e., male or female). This feature might be a good enrollment predictor if the population at UNM tends to lean towards one sex more than the other one.
- ETHNICITY: a categorical variable indicating the ethnicity of the applicant (i.e., black, white, latino and others). This feature might be a good predictor in case applicants of certain ethnicity has a tendency to enroll at UNM more than others.
- ACT_SCORE, SAT_SCORE: discrete variables reflecting the competence level of the applicant. These variables are represented by the ACT and/or the SAT scores. This feature might be a good predictor since students usually would rather enroll in colleges whose student population has a similar competence level.
- GPA: a continuous variable representing the high school GPA of the applicant. Its value ranges between 0 and 5. Similar to the ACT_SCORE and SAT_SCORE variables, GPA reflects the competence level of the applicant.
- FIRST_GENERATION: a binary variable indicating the education level of the parents. The label is 1 if at least one of the parents went to college and 0 otherwise. Usually parents provide their children with an advice on deciding which college to attend. Thus this variable might be a good predictor.
- PARENT_INCOME: a continuous variable indicating the total income of the parents. As mentioned earlier, this feature reflects the socioeconomic status of the parents and should have a major influence on the student's decision choosing which institution to attend.
- STUDENT_INCOME: a continuous variable indicating the income of the student in case he or she has a job. Similar to the PARENT_INCOME variable, STUDENT_INCOME has an influence on the applicant's decision.
- RESIDENCY_STATE: a categorical variable indicating the residency status of the applicant. This variable is a relative measure of the distance from the applicant's residency to UNM campus. It has four labels: 0 indicating that the applicant resides in New Mexico and thus considered as in-state student; 1 indicating that the applicant resides in either Texas, California, Arizona or Colorado. Applicants in those states are eligible to the Amigo scholarship which allows them to pay in-state tuition if they meet certain criteria; 2 indicating that the applicant is non-resident; 3 indicating that the applicant is international. This variable might be a good predictor since it reveals the type of relation between the distance from the campus (implicitly the cost) and the applicant's decision.
- INSTITUTIONAL_MONEY: a continuous variable indicating the amount of the financial aid assigned to the applicant by the institution (i.e., UNM).
- BRIDGE, SUCCESS: binary variables indicating the type of the financial aid offered by UNM. BRIDGE is a reward given for freshman students in their first semester. It is exclusively given to applicants with certain aptitude levels; SUCCESS is a reward given for freshman students in their first semester. It is eligible to applicants with financial needs.
- FEDERAL_MONEY: a continuous variable indicating the amount of the financial aid assigned to the applicant by the federal government.
- APPL_DECISION_DIFF: a discrete variable indicating the total number of days between the time of the application submission and the time of the admission decision. The gap between these two events might be a good predictor. For example, if the admission decision was taken shortly after the application submission, this might provoke the applicant's tendency to enroll at UNM.
- APPLY_AFEB: a binary variable indicating the month during which the application is submitted. The label is 1 if the application is submitted after February and 0 otherwise.

There are many other features that are used in this work. However we did not mention them all because their characteristics are similar-to an extent- to the above listed features. We believe that the features described above give realistic examples of factors that have the potential to influence the applicant's decision of enrollment.

3. MODELS

In this work we approach the enrollment prediction question from a classification perspective. That is we have a pool of applicants and we want to identify or classify those that are most likely to enroll at UNM. We further divide the classification problem into two main approaches: classification at individual level and classification at a cohort level. The individual level approach predicts the enrollment of an applicant based on a given set of features. Then it determines the total number of enrollment by simply counting the applicants who are predicted to enroll. For this purpose, we use two common machine learning classifiers with two class responses: LR and SVMs. The LR with two class responses is one of the basic classification models that aim to find the relationship between a binary response y and a predictor variable(s) x, which can be in a categorical or numerical scale [6]. The response variable y of the binary logistic regression consists of two categories i.e. success and fail. In some cases, the categories are denoted as 1 for success and 0 for fail. In our case the label is 1 if the applicant is predicted to enroll and 0 otherwise. However, a disadvantage of logistic regression is that the technique is not able to identify possible nonlinear structures in the data. A good alternative in this case would be SVMs. On the other side the cohort approach predicts the enrollment of a cohort of applicants based on a given set of features. Using this approach we directly determine the portion of the applicants' pool that would enroll without identifying them individually. The sections below explain in more detail the difference between these approaches and the models used to implement them.

Variable selection

In this work we use a total of 60 features to predict enrollment. It is always possible that some of these features are redundant or irrelevant. Thus they can be removed from the prediction models without incurring much loss of information. One approach to encounter such features is variable selection. The objective of variable selection is three-fold: improving the prediction performance of the predictors, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data [4]. The variable selection methods are typically presented in three classes: filter method, wrapper method and embedded method. In this work we implement the wrapper methodology which come in two flavors: forward selection and backward elimination. In forward selection, variables are continuously added into bigger and bigger subsets, whereas in backward elimination we start with a set containing all the variables and iteratively remove the variable with the least predictability. Both methods yield nested subsets of variables.

3.1 Classification at cohort level

The LR and SMVs models used in this work predict enrollment at individual level. That is, given a set of features for an applicant a_i, the LR and SVMs models predict the probability of enrollment of a_i (LR) or alternatively give a 0/1 flag indicating if a_i will enroll or not (SVMs). In this work we present a new approach in which we predict enrollment at cohort level. That is, given a set of features for a cohort of applicants c_i, we predict the portion of c_i that would enroll. The main concepts underlying this approach are probability and time series analysis. In the probabilistic approach we define a probability distribution over the features of c_i and accordingly compute the respective portion that would enroll. The results shown by this approach prove to be promising.

The probabilistic model is based on a semi-supervised learning method. It is defined as following:

$$p_X(x) = P_0.p_{X|0}(x) + P_1.p_{X|1}(x) \tag{1}$$

where

$p_{X|0}$ is the distribution over the features of applicants that do not enroll. It is estimated from the training data.

$p_{X|1}$ is the distribution over the features of applicants that do enroll. It is estimated from the training data.

$P_1 = 1 - P_0$ is the fraction of applicants that do enroll.

p_X is the distribution over the features of unlabeled applicant data.

So if the total number of applicants is n then the predicted number that would enroll is $n_{enroll} = nP_1$. In this case all what we need to do is to estimate P_1. Solving (1) for P_1 (using $P_1 = 1 - P_0$) gives:

$$P_1 = \frac{p_X(x) - p_{X|0}(x)}{p_{X|1}(x) - p_{X|0}(x)} \tag{2}$$

true for all x.

The second approach used in this work to predict enrollment at cohort level is based on time series analysis. A time series is a sequence of observations collected over time. Usually these observations are taken at constant intervals (i.e., daily, monthly, annually, etc.). The main object of time series analysis is to reveal the model underlying the process generating the series data. Such a model is used to describe the patterns in the series (i.e., trend , seasonality), explain how past observations influence future ones, and accordingly forecast future values of the series [1]. In this work we use a seasonal autoregressive integrated moving average ($ARIMA$) model to forecast the number of applicants that would enroll at UNM. A seasonal $ARIMA$ model is defined as an $ARIMA(p, d, q)$x$(P, D, Q)_m$ model, where

- p is the number of autoregressive terms
- d is the number of differences
- q is the number of moving average terms
- P is the number of seasonal autoregressive terms
- D is the number of seasonal differences
- Q is the number of seasonal moving average terms
- m is the number of periods per season

4. EXPERIMENTAL RESULTS

In an attempt to empirically validate the performance of our proposed models, we analyzed actual university data. For this purpose we used the data of 54,692 First Time Full Time (FTFT) students who were admittted to UNM between years 2009 and 2016. We used this data set in our work in order to layout the needed features, train our models and test their performance.

4.1 Data-preprocessing

In order to get more consistent and discipline results it is essential to preprocess the data set. For this purpose, we implemented a number of common preprocessing techniques used in machine learning.

For various reasons, the data set used in this work contains missing values. That is for some admitted students, UNM does not have all the required information (ex. parents income). This leaves the values of some features in our data set blank. A basic strategy to overcome this problem is to implement imputation methods such as the mean, median or mode of the row or column in which the missing values are located. Another strategy would be simply to discard or remove the rows and/or columns containing missing values. This might come at the price of losing information. However, this might not be the case if the training data set is big enough in which removing some rows will not impact the model performance. In this work we simply discarded the rows with missing values. Consequently, we were left with a data set of 37,500 student which is enough to train our models. Next we standardized the continuous and discrete features of the data set. We removed the mean value of those features and scale them by their respective standard deviation values. Standardization improve the performance of the models by adjusting the features to the same scale. We also converted categorical features to binary features using one-hot encoding. This estimator transforms each categorical feature with m possible values into m binary features, with only one active.

4.2 Numerical results

We used 60 features to train the LR and the SVMs models. We used 10-fold cross validation to examine the performance of these models and compare their results as well. The performance accuracy of both models is presented in Table 1.

	Performance Accuracy (%)
LR	89.41
SVMs	91.25

Table 1: The performance accuracy of the LR and SVMs models using 10-fold cross validation.

It is important to mention that in our training data set the number of observations in each class is not equal. The number of applicants who enroll at UNM is relatively higher than those who do not enroll. In this case the performance accuracy of the classifier can be misleading. A better metric to test the performance of a classifier is a confusion matrix. It is a technique for summarizing the performance of a classification algorithm. A confusion matrix gives a better idea of what the classification model is getting right and what types of errors it is making. Table 2 and Table 3 show the confusion matrices for the LR model and the SVMs model.

The precision and recall values for the LR and the SVMs models are shown in Fig. 1. Both precision and recall are good measures to examine the relevance of the predicted instances to the actual ones. They are calculated using the confusion matrices and hence they are reliable measures to summarize such matrices.

		Prediction outcome		
		p	**n**	**total**
actual value	**p′**	2060	267	2327
	n′	130	1293	1423
	total	2190	1560	

Table 2: The confusion matrix of the LR model.

		Prediction outcome		
		p	**n**	**total**
actual value	**p′**	2063	201	2265
	n′	127	1359	1485
	total	2190	1560	

Table 3: The confusion matrix of the SVMs model.

The performance accuracy, the confusion matrices and the precision and recall scores of the LR and the SVMs models are very similar. Thus the advantage of the SVMs model over the LR model in identifying nonlinear structures is not utilized here. This suggests that the LR model is sufficient to achieve enrollment prediction with a reliable performance. In this context we applied the LR model again to find a subset of the features that attain a similar performance accuracy without losing much information. So we implemented the forward and the backward variable selection models to remove possible redundant and irrelevant features. The results are shown in Fig. 2. The figure presents the classification error of the LR model using 5-fold cross validation for both backward and forward methods. It shows that the forward method has slightly better performance over the backward method. In fact using a subset of 14 features only (the red circle) can achieve a performance accuracy of 89%. This result is almost equal to the performance accuracy of the LR model when using all the features (Table 1). In other words, we can use these 14 features only to predict the enrollment for any future pool of applicants without the need to include the rest of the features in our prediction models. We provide a description for 10 of these features. The description is provided below. These features are sorted according to the predictive importance criterion proposed by the forward selection method. They are listed in a descending order:

- STATE_AWARD_ORIGINAL: This is a continuous variable. It is the amount of the scholarship offered to the

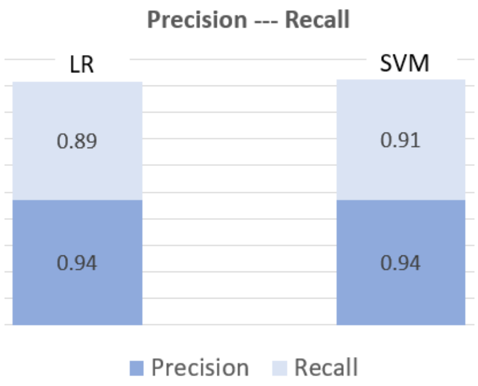


Figure 1: The precision and recall for the LR and SVMs models.

applicants by the state of New Mexico. Perhaps the most important among others is the lottery scholarship. The results presented by our LR model show that on average applicants with state awards tend to enroll more at UNM.

- FIRST_DECISION_AFEB: This is a binary variable. It represents the time of the admission decision. The label is 1 if the admission decision is taken after February (i.e., March, April, May, June and July) and 0 otherwise. The results show that applicants have more tendency to enroll at UNM if the admission decision is taken after February.

- SUCCESS: This is a binary variable. It is the reward given for applicants in their first semester. It is eligible to those with financial needs. The total amount of the reward is 1,000 \$. The LR model shows that applicants with SUCCESS rewards are more likely to enroll at UNM.

- GPA: This is a continuous variable. It represents the high school GPA of the applicants. The results show that applicants with high school GPA between 3.0 and 3.5 tend to enroll more at UNM compared to other applicants.

- RESIDENCY_STATE: This is a categorical variable indicating the residency status of the applicant. The results show that applicants who resides in NM are more likely to enroll at UNM (not surprising!).

- FAFSA_BDEADLINE: This is a binary variable. It indicates if the applicants submit the Free Application for Federal Student Aid (FAFSA) before the deadline set by UNM. The results show that applicants who submit the FAFSA before the deadline tend to enroll more at UNM compared to those who submit after the deadline.

- LOW_INCOME: This is a binary variable. It reflects the socioeconomic status of the parents. The results show that applicants whose parents have a low income are more likely to enroll at UNM.

- BRIDGE: This is a binary variable. It is the reward given for freshman students in their first semester. It is exclusively given to applicants with certain aptitude levels. The total amount of the reward is 1,500 \$. The LR model shows that applicants with BRIDGE rewards are more likely to enroll at UNM.

- APP_AFEB: This is a binary variable. It represents the time when the applicants submit their applications. The label is 1 if the submission is done after February (i.e., March, April, May) and 0 otherwise. The results show that applicants have more tendency to enroll at UNM if the submission is done after February.

- FED_AWARD_ORIGINAL: This is a continuous variable. It is the amount of the financial aid offered to the applicants by the federal state. Perhaps the most important is the Pell grant. The results presented by our LR model show that on overage applicants with federal awards tend to enroll more at UNM.

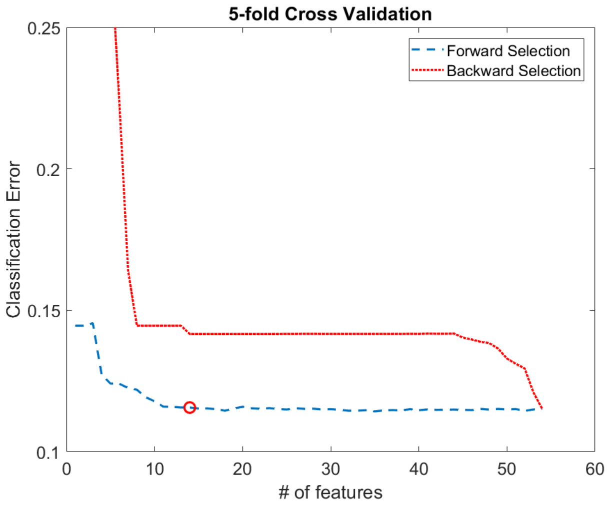


Figure 2: The classification error of the backward and forward variable selection methods implemented using the LR model to predict enrollment.

Cohort prediction

As mentioned earlier the LR and the SVMs models predict enrollment at individual level. In this work we propose alternative approaches where we predict enrollment at cohort level. The first approach is probabilistic in which the total enrollment is computed using (2). We implemented this approach following these set of steps:

- Use previous year data to estimate $p_{X|0}$ and $p_{X|1}$ (labeled data).
- Use current year data to estimate p_X (unlabeled data).
- Use (2) to estimate P_1 for current year.
- Predict the total enrollment: $n_{enroll} = nP_1$

To empirically validate our proposed model we used actual university data for UNM students admitted in years 2015 and 2016. We used the 2015 cohort as a training data set to estimate $p_{X|0}$ and $p_{X|1}$. Then we used the 2016 cohort to estimate p_X and accordingly compute P_1 using (2). The actual and the predicted total enrollment for the 2016 cohort are shown in Table 4.

	Total enrollment (2016)
Actual	3402
Predicted	3478

Table 4: The total enrollment of 2016 cohort at UNM.

We repeated the same procedure, however this time using the 2016 cohort as a training data set to predict the enrollment of the 2015 cohort. The results are shown in Table 5.

	Total enrollment (2015)
Actual	3320
Predicted	3239

Table 5: The total enrollment of 2015 cohort at UNM.

It is essential to mention that we estimated p_X, $p_{X|0}$ and $p_{X|1}$ using only one feature. We evaluated these densities using the histogram method. Then we used (2) to estimate P_1 at multiple x values (histogram bins) and averaged these results to obtain a final P_1 estimate. A remarkable observation using this approach is the accurate predictions using just one feature. This is reasonable. The feature does not have to provide good discrimination because we are not trying to predict individual enrollment; instead we just need to estimate P_1.

Time series analysis is another approach to forecast student enrollment. Unlike the other classification models used in this work, a time series model does not require features to carry out predictions. It only requires a sequence of observations collected over time. This sequence enables us to reveal the model underlying the process generating the series data. In this context we collected the number of students enrolled at UNM for spring, summer and fall semesters of each year. The study contains students enrolled at UNM between years 2003 and 2016. The time series data for this study is shown in Fig. 3 (black color). Note that the number of periods, m, in the series is 3 referring to spring, summer and fall semesters. In this work we used the Akaike Information Critera (AIC) as a statistical measure to choose the $ARIMA$ model that best fits the series. AIC is a widely used measure in statistics. It reflects the robustness of the fitted model in a single value. When comparing two $ARIMA$ models, the one with the lower AIC is generally "better". The parameters of the $ARIMA$ model that best fit the time series of the UNM enrollment data are $p = 0$, $d = 0$, $q = 0$, $P = 1$, $D = 0$ and $Q = 3$ (i.e., $ARIMA(0,0,0)\text{x}(1,0,3)_3$). The fitted model is represented by the red curve in Fig. 3. This model has the lowest AIC and we used it to predict the enrollment at UNM for spring, summer and fall semesters of the 2017 cohort. The predicted numbers are represented by the blue curve of Fig. 3. Table 6 shows the actual versus the predicted enrollment numbers at UNM for the 2017 cohort with 80% confidence interval.

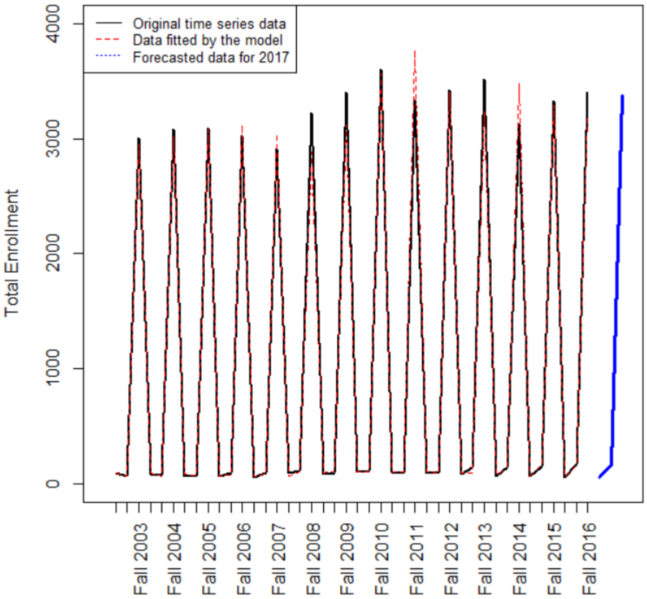


Figure 3: The $ARIMA(0,0,0)\text{x}(1,0,3)_3$ model for the enrollment data at UNM.

	Actual total enrollment (2017)	Predicted total enrollment (2017)	Lower bound (80%)	Upper bound (80%)
Spring	70	57	48	68
Summer	132	153	129	181
Fall	3219	3380	2850	4010

Table 6: The actual and the predicted enrollment of the 2017 cohort at UNM.

5. CONCLUSION

In this paper we shed the light on factors that influence the enrollment decision of applicants. We use machine learning methods to measure the level of correlation between enrollment and such factors. In particular we approach the enrollment prediction question from a classification perspective where we need to identify the likelihood of enrollment for a pool of applicants. We further divide the classification problem into two main approaches: classification at individual level and classification at a cohort level. The individual level approach predicts the enrollment of an applicant based on a given set of features. Then it determines the total number of enrollment by simply counting the applicants who are predicted to enroll. For this approach we implemented a LR model and an SVM model. On the other side the cohort approach predicts the enrollment of a cohort of applicants based on a given set of features. For this approach we implement a semi-supervised probability model and a time series model. Using this approach we directly determine the portion of the applicants' pool that would enroll without identifying them individually. The results show that our proposed models can predict enrollment with reliable accuracy using only a small set of features related to student and college characteristics.

6. REFERENCES

[1] Applied time series analysis. Penn State Eberly College of Science. Available at `https://onlinecourses.science.psu.edu/stat510/node/47`.

[2] Office of institutional analytics. Available at `http://oia.unm.edu/facts-and-figures/freshman-cohort-tracking-reports.html`. Accessed: 1-2-2018.

[3] Enrollment management in higher education - defining enrollment management, key offices and tasks in enrollment management, organizational models. Education Encyclopedia - StateUniversity.com, 2013.

[4] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, Mar. 2003.

[5] D. Hossler and . Bean, John P. *The strategic management of college enrollments*. San Francisco, Calif. : Jossey-Bass, 1st ed edition, 1990. Includes bibliographical references (p. 303-318) and index.

[6] H.-A. Park. An introduction to logistic regression: From basic concepts to interpretation with particular attention to nursing domain. *Korean Society of Nursing Science*, 43(2), 2013.

Re-designing the Structure of Online Courses to Empower Educational Data Mining

Zhongzhou Chen
University of Central Florida
4111 Libra Drive
PSB 153
+1 321-236-8568
Zhongzhou.Chen@ucf.edu

Sunbok Lee
University of Houston
3695 Cullen Boulevard
Heyne building 231D
+1 7816459203
slee95@Central.UH.edu

Geoffrey Garrido
University of Central Florida
4111 Libra Drive
PSB 153
+1 3219873375
Geoff.garrido@knights.ucf.edu

ABSTRACT

The amount of information contained in any educational data set is fundamentally constrained by the instructional conditions under which the data are collected. In this study, we show that by re-designing the structure of traditional online courses, we can improve the ability of educational data mining to provide useful information for instructors. This new design, referred to as Online Learning Modules, blends frequent learning assessment as seen in intelligent tutoring systems into the structure of conventional online courses, allowing learning behavior data and learning outcome data to be collected from the same learning module. By applying relatively straightforward clustering analysis to data collected from a sequence of four modules, we are able to gain insight on whether students are spending enough time studying and on the effectiveness of the instructional materials, two questions most instructors ask each day.

Keywords

Online Instructional Design; Clustering Analysis; Data Interpretability; Supporting Teachers

1. INTRODUCTION

The central goal of educational data mining is to “mine educational data sets to answer educational research questions that shed light on the learning process”. To this end, the predominant focus of the EDM community has been on developing and advancing methods and algorithms to effectively extract information from existing educational data sets. However, the amount of information contained in any given data set is fundamentally constrained by the instructional conditions under which the data is collected [17], such as the nature of the learning tasks, the design and organization of instructional contents, and even the available features of the educational platform. As a simple example, if the final exam is the only assessment administered in an online course, then information about students’ content mastery at any other time during the course is obviously not contained in the data. Therefore, we ask the question: is it possible to enhance the ability of EDM to provide useful information for instructors, by re-designing the structure of the online course to improve the quality of the data that it produces?

Many of today’s online courses more or less inherited their structure from their off-line, face-to-face predecessors. For example, many MOOCs are created directly based on existing face-to-face courses [9, 29, 33]. Those courses typically contain a variety of learning resources, from e-text and videos to problems and forums, organized into week-long units. This structure allows students to display a plethora of different learning behaviors, which has become the focus of many recent studies in EDM. [2, 14, 20, 24, 27]

On the other hand, students’ learning outcome is assessed relatively sparsely in a typical online course. Many recent studies still use “certification rate” or “retention rate” as a proxy for learning over the entire course [14, 21, 27], which can be problematic [19]. Moreover, very few online courses contain any form of pre-test [12],. This is particularly problematic for learning measurement in MOOCs, as there are significant variations in students’ incoming knowledge and background [11, 19]. Insufficient assessment of learning outcome made it difficult for researchers to make meaningful correlations between learning behavior and learning outcome.

In contrast, students’ knowledge state is being constantly assessed in intelligent tutoring systems (ITS), another online instructional system widely studied by the EDM community [4, 15, 18, 30]. A number of methods have been developed to measure students’ learning progress in a ITS with high resolution [13, 22, 23]. However, students’ learning behavior is much more restricted in many ITS as compared to online courses, and oftentimes instructional materials in a ITS consist of only simple hints or feedback texts.

Can we re-design the structure of online courses to include certain features of ITS so that it contains more frequent and accurate learning assessment, while still providing enough freedom for students to display a variety of learning behavior? In this paper we present such an attempt at combining the advantages of both systems, by constructing a small online course consisting of a sequence of four Online Learning Modules (OLMs). Each module contains both instruction and assessment, which enables us to make correlated measurements on students’ learning behavior and learning outcome in close proximity. Moreover, students are required to make one attempt on the assessment before accessing the instruction, which serves as a de-facto pre-test for each learning module. We demonstrate that by applying relatively simple data mining algorithms, data produced by OLMs could provide valuable insight on two questions that every instructor encounters on a daily basis: Q1: Are students spending enough time and effort studying the materials? Q2: How effective are the instructional resources in the course?

Both questions are best answered when considering learning behavior and learning outcomes together. For Q1, "enough time" is best defined for a given instructional resource when students spending less than that time have poorer learning outcomes; For Q2, "effectiveness" can be more accurately measured from the learning outcome of students who spent adequate time and effort learning from the resources. In the remainder of this paper, we will first introduce the design of OLMs and implementation of the current study, then describe the data collection procedure, analysis and visualization methods, followed by the outcomes of the study and ending with a discussion of the impact of this study on potential future research.

2. METHODS

2.1 Design of OLMs

The design of OLM is inspired by research on deliberate practice [16] and mastery-based learning [7, 8], and in particular influenced by the design of the ASSISTMENTs tutoring platform [3, 18]. Each OLM module contains an instructional component (IC) and an assessment component (AC) (Figure 1). The IC consists of both instructional text and ungraded practice problems separated into multiple pages, focused on teaching a single physics concept or a problem-solving skill. Students receive immediate feedback and have access to the problem solution after attempting any practice problem. Each IC typically takes about 10 minutes to an hour for a student to finish, which resembles a small unit in an online course. The AC consists of either 2-3 simple multiple-choice concept problems or 1 complex multiple-choice problem, depending on the focus of the module.

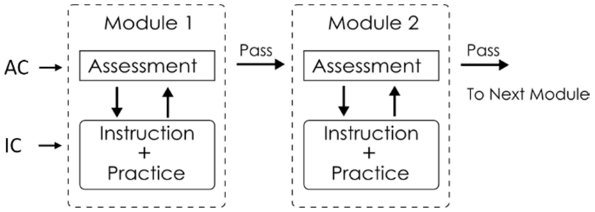


Figure 1: Schematic representation of the structure of OLM and OLM sequence

A series of OLMs are combined sequentially to form a learning unit on a given topic. A student passes a module by correctly answering all the questions in the assessment component, and can proceed onto the next module only after passing the current one. Each student can have multiple attempts on the AC. On each new attempt, a slightly different version of the assessment problem(s) drawn from a problem bank is presented to the student.

A key feature of OLM is that students are required to make at least one attempt on the assessment before being given access to the IC. After the initial attempt, students can either study the IC, or make additional attempts on the assessment. On each attempt the student is presented with a slightly different problem until the problem bank in the assessment component is depleted. During an attempt the IC is temporarily locked from access.

The OLM design has three major advantages for data collection and analysis: First, students' AC attempts before and after instruction serve as de-facto pre and post-tests, increasing the accuracy and frequency of learning measurement. Second, the length and types of learning resources in the IC allows for a richer variety of student learning behavior to be observed compared to many ITS. Finally, by combining instruction and assessment into one module, it allows for observations of learning outcome and learning behavior to be interpreted in the context of each other.

2.2 Study Design and Data Collection

Individual OLMs were created on the award-winning learning objects platform, Obojobo, developed by the Learning System and Technology (LS&T) team at the Center for Distributed Learning at University of Central Florida [6], and administered to students as a sequence via the Canvas learning management system. For the current study, student subjects were recruited from three sections of calculus-based college introductory physics course at University of Central Florida during the Spring 2017 semester. The OLMs were provided to students as an optional reviewing tool for an upcoming exam.

Four OLMs were created on the topic of conservation of mechanical energy with each module focusing on a single concept or a problem-solving skill. The problem bank of each AC contains 3 isomorphic multiple-choice problems authored based on published assessment instruments in physics[32]. The distractors in each problem are designed to capture common student misconceptions.

The number of students who made at least 1 attempt on the AC of modules 1-4 are 75, 54, 47 and 40 respectively. In this study, students were allowed 50 attempts on each module to ensure that they can all proceed to the next module.

Time-stamp data on the following types of student events are collected by the Obojobo platform: Entering and exiting a page in both IC and AC; Starting and finishing an attempt on either an assessment problem or a practice problem; Viewing a practice or assessment problem; Submitting an answer to a practice or assessment problem; Outcome of each attempt at the AC.

2.3 Data Analysis

2.3.1 Capturing Learning Behavior within Longest Study Session

All of the interactions by one student with the IC that took place between two consecutive assessment attempts are treated as a single "study session" (SS). A student can have multiple SS by going back and forth between the IC and the assessment component. For answering the questions in this manuscript, we only consider SS that took place before the first time a student passes the assessment component is recorded.

In a total of 168 occasions where a student interacted with the IC of a module, 76% (127) of the time all interactions took place in a single SS. In most of the other occasions, there is a major SS that is significantly longer than the other SS. In only 4 cases did the second longest SS reach at least 50% as long as the longest SS (LSS). Since the majority of students' learning behavior for each module took place during their LSS, it serves as a good approximation for measuring students' learning effort of the given module. For the current analysis, students' learning behavior within the LSS is characterized along three dimensions:

1. The duration of the LSS, measured as the sum of the times spent on each accessed page in the IC.

2. The average number of attempts made on practice problems, measured as the total attempts made divided by the number of practice problems viewed by the student.

3. The percentage of contents accessed, measured as the sum of page entering events plus problem viewing events, divided by the sum of the number of pages and the number of practice problems in each module.

2.3.2 Clustering Analysis of students' learning behavior

In this study, we assume that students' learning behavior will form multiple clusters due to different learning strategies, habits and incoming knowledge states. In order to identify such subgroup, we used a mixture model in which the whole population distribution is represented by the sum of component distributions representing subgroups, and the probabilities of students' belonging to subgroups or classes are estimated. We used Mplus software [28]to fit the mixture model to our data. The optimal number of classes was judged based on six statistical indices provided by Mplus: Akaike Information Criterion (AIC)[1], Bayesian Information Criterion (BIC)[31], Sample-size Adjusted BIC, Vuong-Lo-Mendell-Rubin Likelihood Ratio (VLMRLR) Test[34], Lo-Mendell-Rubin Adjusted LRT (LMRALRT) test[25], and Bootstrap Likelihood Ratio (BLR) test[26]. AIC, BIC, and sBIC are goodness of fit indices which consist of -2 log (likelihood) and an additional term for penalizing a complex model. Each tries to strike balance between fit (-2 log (likelihood)) and parsimony (a penalty term), and a smaller value indicates better fit. The other three indices are the statistical tests comparing how well the data is fitted by models with n and n-1 classes, i.e. p-value less than .05 from those tests indicates that the current model with n-classes has a significantly better fit than the model with (n-1)-classes. In short, the optimal number of classes can be determined by running mixture models with a different number of classes (e.g., models with 1,2,3, and 4 classes) and by selecting the model showing the overall best fit to the data based on those six indices.

2.3.3 Categorizing Learning Outcome

Students' learning outcome from each module can be classified into four classes according to performance in the AC and time of attempt relative to the LSS:

1. **Initial Pass (InitP):** Passing the AC within 2 attempts before LSS. Those students did not need to learn from the IC, although a small fraction still interacted with the IC. An earlier study on students' test-taking effort on the initial attempt estimated that 80-85% of the students took the attempt seriously. [10]

2. **Effective (Eff):** Passing the AC within 2 attempts after LSS (not including attempts before LSS).

3. **Ineffective (Ineff):** Passing the AC using more than 2 attempts after LSS.

4. **Abort:** Never passing the AC, thus cannot access the next module in the sequence.

In addition, in a few cases a student passes the AC using more than 2 attempts without accessing the IC. Since those students are more likely to be randomly guessing the answer rather than actually doing the problem, we also categorized them as "Abort".

3. RESULTS AND DISCUSSION

3.1 Results

3.1.1 Identifying Clusters of Learning Behavior

Cluster Analysis was performed on all three dimensions of learning behavior for each module, for all students who didn't pass the module on their attempt before LSS. The 3-dimensional clustering analysis did not converge for any module likely due to small sample size. Clustering analysis on both average number of attempts and percentage of content accessed always favored single cluster for every module. The mean average number of practice problem attempts are between 1 and 3 attempts for all modules, and the mean content accessed is more than 95% for all modules.

For the time-on-task dimension, initial clustering results were significantly distorted by a few data points with extremely long and scattered LSS durations, most likely due to students leaving their computer without logging off of the system or idling. Thus, clusters with less than 5 students and significantly larger mean values were removed and the clustering analysis re-run, until no such cluster existed. We also found a small cluster of students with mean LSS time of 30 seconds and interacted with the IC of Module 1 only. Those students were also removed since they are likely students who are curious about the new system but did not seriously study the content. The resulting statistical indices for different number of clusters are listed in TABLE 1.

TABLE 1: Statistical indices of mixture-model clustering analysis. Favorable values are highlighted in red.

Module	class	AIC	BIC	sBIC	VLMR (p)	LMR (p)	BLRT (p)
Module 1 (N = 36)	1	38.6	41.8	35.5	NA	NA	NA
	2	37.4	45.3	29.7	0.05	0.07	1.00
	3	37.3	50.0	25.0	0.08	0.11	0.43
	4	did not converge					
Module 2 (N = 38)	1	100.4	103.6	96.4	NA	NA	NA
	2	88.6	96.8	81.2	0.01	0.01	0.00
	3	90.8	103.9	78.9	0.56	0.58	1.00
	4	90.0	108.0	73.6	1.00	1.00	1.00
Module 3 (N = 37)	1	119.4	122.7	116.4	NA	NA	NA
	2	116.3	124.3	108.7	0.02	0.03	0.15
	3	116.6	129.5	104.5	0.02	0.03	1.00
	4	116.8	134.5	100.1	0.27	0.30	1.00
Module 4 (N = 26)	1	95.2	97.7	91.5	NA	NA	NA
	2	90.1	96.4	80.9	0.01	0.01	1.00
	3	88.2	98.3	73.4	0.05	0.06	1.00
	4	88.5	102.3	68.2	0.14	0.17	1.00

For all four modules, a 2-cluster model is either most favorable, or equally as favorable as a 3-cluster model. Therefore, we adopt a 2-cluster model for the LSS duration dimension for each module, referring to the cluster with shorter mean time as "Brief" and the longer mean time as "Extensive" (Ext). One possible interpretation is that the "Brief" clusters consist of students who had some level of initial understanding and needed a quick refresh of the content knowledge, while the "Extensive" clusters are students who failed to learn the content properly during regular lecture, and are actually learning from the IC of the modules.

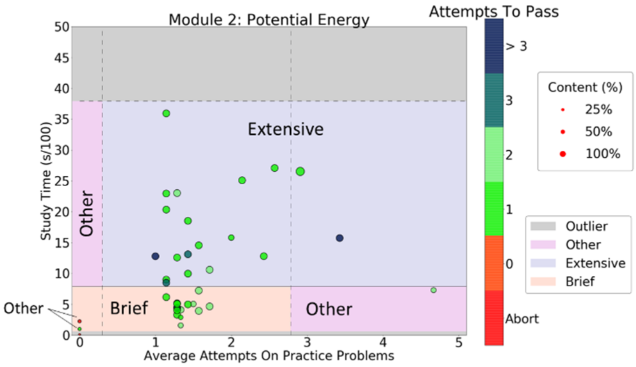


Figure 2: Example of refinement of clustering analysis outcome: Horizontal solid line divides the two clusters. The vertical dashed line indicates 1.5 standard deviation from population mean.

The clusters are further refined by labeling a few students who displayed inconsistent behavior along the other two dimensions as "other". As illustrated in Figure 2 , a student in the "brief" cluster who makes significantly more attempts on practice problem (more than 1.5 sd above the group mean) is labeled as "other" since his/her learning behavior is very different from other students (lower right purple area). Similarly, any student in the "Extensive" cluster whose average practice problem attempt or percentage of content accessed is 1.5 sd less than the population mean is also labeled as "other", since the student is most likely not meaningfully engaged with the learning material. Finally, a few students who didn't attempt any practice problems, and/or interacted with the IC for less than 60 seconds are also labeled as "other" as their learning behavior is significantly different from the rest of the population.

The refinement strategy is illustrated in Figure 2 using data from Module 2 as an example. The final clusters of students' learning behavior are listed in TABLE 2.

TABLE 2: Refined learning behavior clusters

Modules	Cluster	N	mean (s)	Var. (s)
1	Brief	25	519	75
	Ext	8	1272	12
	Other	3	NA	NA
2	Brief	15	416	36
	Ext	19	1594	675
	Other	4	NA	NA
3	Brief	18	1233	495
	Ext	16	3382	165
	Other	3	NA	NA
4	Brief	18	1141	576
	Ext	5	4175	256
	Other	3	NA	NA

3.1.2 Combining Learning Behavior with Learning Outcome

To visually represent the relation between learning behavior and learning outcome in each module, we plot both types of information together in four sunburst charts shown in Figure 3. The inner rings show the distribution of four classes of learning outcome, while the outer ring shows the distribution of the three learning behavior clusters within each learning outcome classes. Some of the key observations from the data are summarized in TABLE 3.

Looking at assessment performance alone, Modules 3 and 4 are significantly harder than modules 1 and 2, judging by both the fraction of students in InitP (Fisher's exact test, $p = 0.01$) and the total fraction of students who passed the module either before or after accessing the IC (Tot.Pass) ($p < 0.01, \chi^2 = 40, df = 3$). The total number of passing students is the sum of the InitP group and the Eff group.

A noteworthy observation is that initially less students passed module 2 than module 1, but after studying the IC the trend was reversed.

The effectiveness of the IC can be estimated by the ratio of the size of Eff vs. Ineff classes. For simplicity, in the remainder of the paper (including TABLE 3) we will include the students in the "Abort" class into the "Ineff" class, which now contain all students who failed to pass within two attempts after LSS. Modules 1 and 2 have a significantly higher ratio of Eff vs. Ineff ($p < 0.01, \chi^2 = 34.39, df = 3$). (Test still significant when either module 2 or module 4 is excluded).

From the learning behavior perspective, Modules 2 and 3 have significantly higher Ext vs. Brief ratio ($p = 0.01, \chi^2 = 11, df = 3$) as compared to the other two modules. Somewhat unexpectedly, the size of "Extensive" group in module 4 is the smallest of the four, consisting of only 5 students.

TABLE 3: Main observations. The total number includes students who passed the AC before studying IC

Modules	N	InitP	Tot. Pass	Eff/Ineff	Ext/Brief
1	47	0.26	0.79	2.50	0.32
2	40	0.12	0.88	6.00	1.27
3	35	0.03	0.57	1.27	0.89
4	25	0.04	0.16	0.14	0.28

Finally, the correlation between the learning behavior clusters ("Brief", "Extensive") and the learning outcome measures ("Eff", "Ineff") are not significant when the four modules are combined (Fisher's exact test, $p = 0.35$, $\mathrm{OR} = 0.65$). This correlation is also not statistically significant at $p = 0.05$ level when each of the four modules were tested individually. In other words, there is no significant difference in the probability of passing each module after learning from the IC between the "Brief" and "Extensive" groups.

Of the 61 students that are not excluded as an outlier in at least one of the modules, only 4 are Brief and Ineff (including Abort) for 2 modules, and no student is both Brief and Ineffective for more than 2 modules. In comparison, 3 students are Extensive and Effective for 3 out of 4 modules.

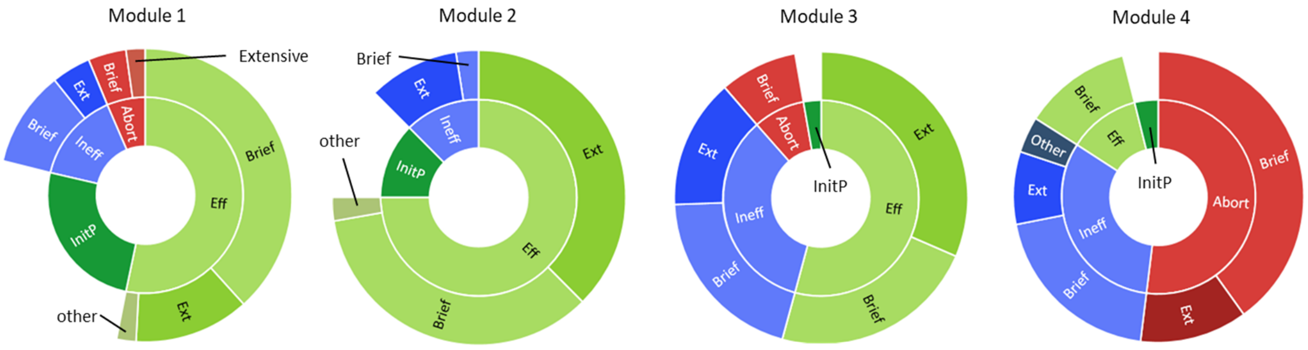


Figure 3: Sunburst charts representing students' learning behavior and learning outcome

3.2 Discussion

By combining learning behavior measurement with learning outcome measurement, we are able to answer both research questions introduced in Section 1 and provide useful information for instructors regarding the four OLMs. For RQ1, data suggests that students in this study are consciously adjusting their learning effort according to their own learning needs and the difficulty of the task. This claim is supported by the lack of correlation between the two learning effort clusters and the three learning outcome clusters, together with the fact that only a few students were consistently "Brief and Ineffective" or "Brief and Abort". In other words, all of the students can be viewed as spending "enough time" on the IC, as there are no clear benefits associated spending longer time. At least, the instructor should be advised to only give the suggestion of "study harder" to the 4 students who are "Brief" and "Ineffective" for 2 out of 4 modules. Had we only considered behavior measurement alone, many more students who have better incoming knowledge on the topic would have been misclassified as less motivated.

One possible explanation for this observation is that since this is a voluntary, not-for-credit activity, only motivated students attempted the OLMs. In future studies it will be interesting to see if the outcome changes when OLMs are being assigned for credit to the entire class.

Our data analysis also provides rich information with regard to the quality of learning resources in the OLMs (RQ2). Among the four modules, Module 1 is the easiest, with high initial passing rate and low "Extensive" vs. "Brief" ratio, suggesting that many students only needed a quick "refresh" of the content. The assessment of Module 2 is slightly harder (lower fraction of InitP), but most students were able to successfully learn the content by carefully studying the IC, as indicated by significantly higher Eff to Ineff ratio and the highest Extensive to Brief ratio. These data suggest that the resources in the IC of Module 2 are effective for the current student population. Note that if only a posttest were given in this course, we might have concluded that problems in modules 2 were easier than those in module 1 without considering students' prior knowledge and learning effort. The AC of Module 3 is even harder, and despite a significant fraction of students in the "Extensive" cluster, a smaller fraction of students passed the AC after studying the IC, suggesting that the instructional resources in the IC of module 3 are less effective and need more improvement.

Module 4 has an unusually large fraction of "Abort" students, and a surprisingly small "Ext" cluster despite being the hardest of all four modules. A likely explanation is that many weaker students find this module too challenging, and lack both the confidence and the incentive to study it as it is the last module in the sequence. In fact, half of the students (9 out of 16) belonging to the "Ext" cluster in Module 3 aborted module 4, whereas only a third (6 out of 18) of students in the "Brief" cluster of Module 3 aborted Module 4.

The majority of the above information is intuitively represented in the sunburst charts in Figure 3, which clearly signals to the instructor that Modules 3 and 4 needs to be improved, and that at least on Module 3, students' lower performance is not caused by insufficient learning effort, but rather ineffective instructional resources.

It is worth pointing out that the mean duration of the "Brief" cluster for modules 3 and 4 are similar to that of the "Ext" cluster for Modules 1 and 2. One possibility is that the learning behavior of the "Brief" cluster in Modules 3 and 4 are more similar to the "Ext" cluster of Modules 1 and 2. However, we only found 4 students who changed from the "Ext" cluster in Module 2 to the "Brief" cluster in Module 3. We think that a more dominant factor is simply that the IC in Modules 3 and 4 contains instructional resources that took longer to go through than Modules 1 and 2. However, examining whether the same cluster across different modules originate from similar learning behavior is an important question for future research.

Finally, we would like to address a couple of detailed choices in both study design and data analysis. First of all, the choice of using 2 attempts instead of one as the threshold for passing a unit is to mitigate the effect of carelessness in students and the possibility of accidentally selecting the wrong choice item. Furthermore, research on multiple attempts has shown that subsequent attempts on problems have equal discrimination power as the initial attempt [5].

Secondly, even though students have already been exposed to the content in lecture, it is clear from the analysis that most of them still need to either refresh or learn the content from the OLMs. We believe that the methods developed in this research are general to most online-courses, especially when we are facing an increasingly diverse student population in higher education and MOOCs in particular.

Finally, choosing mixture-model clustering analysis to capture patterns in students' learning behavior has two major advantages. First, it provides a systematic method to remove outliers in the data, and second, it accommodates the fact that different resources intrinsically require different amounts of time to study, by providing natural cutoffs between "Brief" and "Extensive" clusters.

4. SUMMARY

In this paper, we presented a case where a re-design of the online course structure enabled new methods of data analysis and visualization that provide useful information for instructors. The OLMs are designed to measure both learning behavior and learning outcome in the same module, greatly improving the interpretability of both types of data. Future larger scale studies involving more advanced data mining methods will likely provide insight into even more aspects of students' learning process, such as knowledge transfer, motivation, and meta-cognitive skills.

As data collection and analysis becomes an increasingly important and integrated part of today's technology enhanced education system, it is valuable for data scientists to be more actively involved in the design of instructional systems, resources and environments, rather than simply being on the receiving end of educational data. Design choices that are made to improve the quality of data, even as small as requiring an extra click to view a given problem, may significantly enhance the power of educational data mining, which eventually benefits teaching and learning.

5. ACKNOWLEDGMENTS

We thank the UCF LS&T team led by Dr. Francisca Yonekura for developing the Obojobo platform for implementing OLMs and Dr. Patsy Moskal at the Center for Distributed Learning at UCF for commenting on the manuscript. The project was supported by star-up funds from the University of Central Florida.

6. REFERENCES

[1] Akaike, H. 1974. A New Look at the Statistical Model Identification. 215–222.

[2] An, T.-S. et al. 2017. Can typical behaviors identified in MOOCs be discovered in other courses? *Proceedings of the 10th International Conference on Educational Data*

Mining (2017), 220–225.

[3] Baker, R.S. 2016. Stupid Tutoring Systems, Intelligent Humans. *International Journal of Artificial Intelligence in Education*. 26, 2 (2016), 600–614. DOI:https://doi.org/10.1007/s40593-016-0105-0.

[4] Baker, R.S.J.D. 2010. Data mining for education. *International Encyclopedia of Education*. 7, (2010), 112–118. DOI:https://doi.org/10.4018/978-1-59140-557-3.

[5] Bergner, Y. et al. 2015. Estimation of ability from homework items when there are missing and/or multiple attempts. *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge - LAK '15*. (2015), 118–125. DOI:https://doi.org/10.1145/2723576.2723582.

[6] Bishop, C. et al. 2013. Pilot Study Examining Student Learning Gains Using Online Information Literacy Modules. *Proceedings of the Association of College and Research Libraries (ACRL) Annual Conference* (Indianapolis, Indiatna, 2013), 466–471.

[7] Block, J. and Burns, R. 1976. Mastery learning. *American Educational Research Journal*. 4, (1976), 3–49.

[8] Bloom, B.S. 1968. Learning for Mastery. *UCLA Evaluation Comment*. 1, 2 (1968).

[9] Breslow, L. et al. 2013. Studying learning in the worldwide classroom: Research into edX's first MOOC. *Research & Practice in Assessment*. 8, March 2012 (2013), 13–25. DOI:https://doi.org/10.1007/BF01173772.

[10] Chen, Z. et al. 2018. Designing online learning modules to conduct pre- and post-testing at high frequency. *2017 Physics Education Research Conference Proceedings* (Cincinnati, OH, Jan. 2018), 84–87.

[11] Chen, Z. et al. 2016. Researching for better instructional methods using AB experiments in MOOCs:Results and Challenges. *Research and Practice in Technology Enhanced Learning*. 11, 9 (Dec. 2016). DOI:https://doi.org/10.1186/s41039-016-0034-4.

[12] Chudzicki, C. et al. 2015. Validating the pre/post-test in a MOOC environment. *2015 Physics Education Research Conference Proceedings*. (2015), 83–86. DOI:https://doi.org/10.1119/perc.2015.pr.016.

[13] Cook, J. et al. 2017. Task and Timing: Separating Procedural and Tactical Knowledge in Student Models. *Proceedings of the 10th International Conference on Educational Data Mining* (2017), 186–191.

[14] Davis, D. et al. 2016. Gauging MOOC Learners' Adherence to the Designed Learning Path. *Proceedings of the 9th International Conference on Educational Data Mining*. (2016), 54–61.

[15] Doroudi, S. et al. 2016. Sequence matters, but how exactly? Towards a workflow for evaluating activity sequences from data. *Proceedings of the 9th International Conference on Educational Data Mining* (2016), 70–77.

[16] Ericsson, K.A. et al. 1993. The Role of Deliberate Practice in the Acquisition of Expert Performance. *Psychological Review*. 100, 3 (1993), 363–406.

[17] Gašević, D. et al. 2015. Let ' s not forget : Learning analytics are about learning. *TechTrends*. 59, 1 (2015), 64–71. DOI:https://doi.org/10.1007/s11528-014-0822-x.

[18] 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 (2014), 470–497. DOI:https://doi.org/10.1007/s40593-014-0024-x.

[19] Ho, A.D. et al. 2014. HarvardX and MITx: The First Year of Open Online Courses, Fall 2012-Summer 2013. *SSRN Electronic Journal*. 1 (2014), 1–33. DOI:https://doi.org/10.2139/ssrn.2381263.

[20] Kizilcec, R.F. et al. 2013. Deconstructing Disengagement : Analyzing Learner Subpopulations in Massive Open Online Courses. *Lak '13*. (2013), 10. DOI:https://doi.org/10.1145/2460296.2460330.

[21] Li, Y. et al. 2017. When and who at risk ? Call back at these critical points. *Proceedings of the 10th International Conference on Educational Data Mining* (2017), 168–173.

[22] Liu, R. and Koedinger, K.R. 2017. Towards reliable and valid measurement of individualized student parameters. *Proceedings of the 10th International Conference on Educational Data Mining*. (2017), 135–142.

[23] Liu, R. and Koedinger, K.R. 2015. Variations in learning rate : Student classification based on systematic residual error patterns across practice opportunities. *8th International Conference on Educational Data Mining* (2015).

[24] Liu, Z. et al. 2016. MOOC Learner Behaviors by Country and Culture; an Exploratory Analysis. *Proceedings of the 9th International Conference on Educational Data Mining* (2016), 127–134.

[25] Lo, Y. et al. Testing the Number of Components in a Normal Mixture. *Biometrika*. Oxford University PressBiometrika Trust.

[26] McLachlan, G.J. 1987. On Bootstrapping the Likelihood Ratio Test Stastistic for the Number of Components in a Normal Mixture. *Applied Statistics*. 36, 3 (1987), 318. DOI:https://doi.org/10.2307/2347790.

[27] Miyamoto, Y.R. et al. 2015. Beyond time-on-task: The relationship between spaced study and certification in MOOCs. *Journal of Learning Analytics*. 2, 2 (Dec. 2015), 47–69. DOI:https://doi.org/10.18608/jla.2015.22.5.

[28] Muthén, L.K. and Muthén, B.O. *Mplus User's Guide Seventh Ediction*. Muthén & Muthén.

[29] Rayyan, S. et al. 2016. A MOOC based on blended pedagogy. *Journal of Computer Assisted Learning*. 32, 3 (2016), 190–201. DOI:https://doi.org/10.1111/jcal.12126.

[30] Romero, C. and Ventura, S. 2010. Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*. 40, 6 (2010), 601–618. DOI:https://doi.org/10.1109/TSMCC.2010.2053532.

[31] Schwarz, G. Estimating the Dimension of a Model. *The Annals of Statistics*. Institute of Mathematical Statistics.

[32] Singh, C. and Rosengrant, D. 2003. Multiple-choice test of energy and momentum concepts. *American Journal of Physics*. 71, 6 (2003), 607.

DOI:https://doi.org/10.1119/1.1571832.

[33] Toven-Lindsey, B. et al. 2015. Virtually unlimited classrooms: Pedagogical practices in massive open online courses. *Internet and Higher Education*. 24, (2015), 1–12. DOI:https://doi.org/10.1016/j.iheduc.2014.07.001.

[34] Vuong, Q.H. 1989. Likelihood Ratio Tests for Model Selection and Non-Nested Hypotheses. *Econometrica*. 57, 2 (Mar. 1989), 307. DOI:https://doi.org/10.2307/1912557.

Using Student Logs to Build Bayesian Models of Student Knowledge and Skills

Huy Nguyen
Lafayette College
nguyenha@lafayette.edu

Chun Wai Liew
Lafayette College
liewc@lafayette.edu

ABSTRACT

Recent works on Intelligent Tutoring Systems have focused on more complicated knowledge domains, which pose challenges in automated assessment of student performance. In particular, while the system can log every user action and keep track of the student's solution state, it is unable to determine the hidden intermediate steps leading to such state or what the student is trying to achieve. In this paper, we show that this information can be acquired through data mining, along with the type, frequency and context of errors that students made. Our technique has been implemented as part of the student model in a tutor that teaches red-black trees. The system was evaluated on three semesters of student data. Analysis of the results shows that the proposed framework of error analysis can help the system in predicting student performance with good accuracy and the instructor in determining difficulties that students encounter, both individually and collectively as a class.

Keywords

Data structures, Bayesian Learning, Error analysis

1. INTRODUCTION

An important goal in assessing student performance is to find out why the student makes certain errors, as it helps the instructor adapt the teaching style/materials accordingly to address the cause of such errors. With the rise of educational technology, it is now often the tutoring system that performs grading tasks in place of the instructor, thereby raising the need for an automated error analysis mechanism. Traditionally, tutoring exercises are often designed as multiple choice questions, where there is a single correct option, while the incorrect options are each worded in a way that targets a specific misconception (e.g., [6]). In this case, knowing which option a student picked is sufficient to infer why she made that decision. However, multiple-choice questions can be answered by pure guessing, and the options presented might not capture the full space of misconceptions that students have, especially if each decision is not a simple primitive choice. Furthermore, recent development of tutoring systems has moved on to more complicated knowledge domains, such as protein folding [2], programming language [7], and database [19]. These domains require students to engage in high level problem-solving tasks instead of simple multiple-choice and short-answer questions. In turn, they also pose challenges to the tutoring system in assessing student performance, namely (1) recognizing when the student is correct, (2) identifying the analyzing the errors made, and (3) predicting when a previous error might occur again.

Tree data structure exercises are an example of problems where the steps are best input graphically to show how the data structure is transformed at each step. Conventional question formats such as multiple-choice would therefore greatly constrain the student's answer and allow for the possibility of guessing. An ideal input mechanism, in this case, should allow the student to freely and easily specify the tree structure, reveal no clue or bias about the solution, and support automated assessment of student answer. In other words, it is to closely resemble a paper exam where students construct their answers from scratch.

The solution to an insertion/deletion tree problem is a sequence of transformations (steps) to be applied to the initial tree; alternatively, it can also be viewed as a list of trees, each resulting from applying a transformation to the tree before it. Determining if an answer is correct is straightforward - we simply check that the default solution's final tree matches that of the student's answer. If they differ, however, determining where and how the student made an error is much more difficult. The primary reason is that there can be multiple valid solutions, each with a different partial ordering of the same set of transformations. Furthermore, when unconstrained by the system, students also tend to combine several base (primitive) steps together into a macro-step, in which case their solution sequence can be shorter than the default solution yet still correct. Despite these difficulties, the assessment task plays an important role in both assigning partial credits to test submissions and informing the instructors about difficulties that students are facing, so that necessary interventions may take place.

This paper presents our approach in solving the assessment problem in the domain of red-black tree, a type of self-balancing binary search tree. In particular, based on analysis of the tutoring system's log data and student answers, we have devised a framework to identify and categorize the errors in their problem-solving process. The output is then used to construct a Bayesian student model, which predicts student performance throughout the tutoring session (i.e., whether the student's next answer will be correct or not). We show that categorizing the identified errors by not only their types but also contexts can help the model achieve good accuracy and provide insights into common patterns of problem-solving behavior in our chosen domain. The contexts can be identified by mining the logs and judiciously combining data to identify contexts that might be temporally connected. The mined data can help identify when temporally adjacent contexts affect student decisions and point out non obvious connections.

2. RED-BLACK TREES

A red-black tree is a self-balancing binary search tree with a number of properties which guarantee an $O(\log N)$ height when the tree has N nodes [5]:

1. The nodes of the tree are colored either red or black.
2. The root node is always black.
3. A red node cannot have any red children.
4. Every path from the root to a null node contains the same number of black nodes.

Search in a red-black tree's operation is identical to that in a conventional binary search tree, while insertion and deletion are performed differently. The top-down algorithm to insert or delete a value from a red-black tree starts at the root and, at every iteration, moves down to the next node, which is a child of the current node. At each node, it applies one or more transformation rules; there are six rules used in insertion: *color flip, single rotate, double rotate, insert node, forward*, and *color root black*. Deletion involves another two, *switch value* and *drop rotate*. The role of these transformations is to change the tree in such a way that when the actual insertion (or deletion) is performed at the leaf node, in most cases no subsequent modifications to the tree are needed in order to preserve its properties. Other types of balanced trees also employ a similar approach. In our work we used red-black tree as an exemplar to evaluate our ideas and implementations, but they should be applicable to balanced trees in general.

In a standard curriculum, students learn about red-black trees right after finishing binary search tree, but often struggle because the tree transformations are quite complicated, especially on a medium-sized tree of more than 10 nodes. Furthermore, the insertion and deletion version of the same transformation (for example, *color flip*) operate differently, causing another source of confusion. A previous study on this domain by Liew & Xhakaj [15] found that red-black trees can be taught and learned effectively using a *granularity approach* - students should iteratively break down the problem into three steps of (1) identifying the current node, (2) selecting the applicable transformation, and (3) applying the selected transformation. Our tutoring system also follows the same approach.

3. RELATED WORK

There are well-studied advantages and disadvantages of both multiple-choice and free-response questions [10]. As the domain knowledge gets more complicated, it becomes more difficult to design multiple-choice tests that accurately reflect the student's level of understanding; on the other hand, free-response questions are not scalable because of the need for human graders. In practice, many intelligent tutoring systems opted for the middle ground by using a restricted language such as numerics for student answers. In this way, there is still a large solution space that makes guessing ineffective while the information derived from students' assessment is accurate enough to be used in constructing a student model. For example, physics tutors such as ANDES [4] and OLAE [16] teach college-level Newtonian mechanics by having students identify the forces acting on a physical object and express them in a system of equations.

Several past works have explored automated assessment in complex domains. For example, [3] uses an online judge system for an introductory programming course that is capable of detecting plagiarism and performing efficient, bias-free assessment. [17] constructs an adaptive grading system that can grade multiple and complex computer literacy assignments while being able to "learn" the correct and incorrect responses and add them to the rubric. Combining both human graders and computer graders, [8] introduces a collaboration framework that aims to minimize human effort in the domain of medical case analysis, using supervised machine learning.

Efforts have also been made to output not only a binary result (correct/incorrect) or numerical score, but also to provide reasonable feedback for both the students and the instructors. Many research works in the domain of introductory programming have been following this direction [9, 20, 21]. In other domains, [14] shows that in the PHYSICS-TUTOR system, where students enter algebraic equations as answer, it is possible to check for dimensional correctness and isolate errors by parsing the submitted answers into binary expression trees. Finally, [11] proposes using case-based reasoning to deliver past instructor feedback to new students who are solving a similar problem, which has been adapted in various tutoring systems.

Predicting student performance is one of the primary goals of student modeling. Traditionally, Bayesian network and its variations [1] are often used because of their accuracy and interpretability. This line of technique has been shown to be effective for tutoring systems that have no prior knowledge about their students, such as the ANDES physics tutor. Later on, ITSs are often deployed multiple times in successive semesters, and the data log from past student interactions can be analyzed by data mining techniques to better predict future students' performance. For instance, [12] builds a logistic regression model on the ANDES dataset to correctly identify 70% of the student's performance, while [18] uses pattern classifier and genetic algorithm to improve the tutoring system's prediction accuracy, which helps identifying weak students early on even in large classes.

In the domain of red-black trees, [15] was among the first tutoring systems developed. Its result shows that the granularity approach, which require students to follow explicit small steps, helped significantly improve their performance in insertion exercises. The system was built only for tutoring, while the tests were conducted on paper and evaluated by a human instructor. [13] proposes preliminary results in automating the test environments and grading with an algorithm that can detect the first error made by students in tree insertion questions.

4. THE RED-BLACK TREE TUTOR

Our tutoring system has three sections - the pre-test, the tutor, and the post-test. In the test sections, a typical insertion (deletion) problem for red-black trees involves inserting a sequence of numbers to a starting tree (or deleting from it). Students have to show the state of the tree after every insertion/ deletion; they are also encouraged to show any intermediate states (the trees that are created along the path to the solution). To this end, the test interface displays a "blank" binary tree canvas of 31 empty nodes. The student can click on any node to specify its value and color - submitting a tree is therefore equivalent to entering all of its nodes to the corresponding position in the tree canvas; nodes that are left empty are assumed to be null black nodes. The interface is designed to look like a sheet of paper with blanks to fill in - in this way, we ensure that (1) the tests do not provide any hints or clues as to what the desired answer would be, and (2) the student's answer is always in a format that can be interpreted and analyzed by the system.

In the tutoring section, students perform the same task of inserting to (or deleting from) a starting tree. However, a node-by-node modification of the current tree is not required; instead, students only need to select a node and the transformation to apply at that node from a drop-down list. The tutoring system has a solver module that can generate a solution for any problem and also check the correctness of the student's selection. If it is correct, the system will automatically apply the chosen transformation and update the information shown in the interface; otherwise, a message is displayed to the student indicating that the current selection is incorrect. We chose this approach based on the finding that learners often have difficulty identifying the transformations rather than applying them [15]; students also find the task of repeated application of the transformations tedious and time consuming.

5. PREDICTION OF STUDENT PERFORMANCE

In order for the system to be dynamic (i.e., to generate dynamic exercises that address an individual student's weakness), it needs to have knowledge of what the student knows and does not know at any given time. In the context of our tutor, the system should be able to predict whether the student's next answer is correct, based on her performance so far in the tutoring session and in the pre-test. To our knowledge there has not been any prior work on performance prediction in the domain of binary search tree. Therefore, to get a better sense of how well the student model performs, we implemented and evaluated three approaches.

5.1 Baseline prediction

Every time the student submits an answer in the tutoring session, the system predicts that the answer is correct with a fixed $p = 0.5$ probability. The performance of this method will serve as a baseline to compare with that of the next two methods.

5.2 Bayesian model with error contexts

We first analyze student answers in the pre-test and identify the first error made (if any) each time the student attempts to insert/delete a single node to/from a tree. Besides the type of error - incorrect node selection/incorrect transformation selection/ incorrect transformation application - and its location - how far did the student progress when the error was made - we are also interested in its context. In insertion exercises, an error context is the subtree surrounding the node at which the error occurred, which includes its parent and two children. In deletion exercises, the context subtree also contains the node's sibling and sibling's children. These definitions were devised based on the knowledge that (1) the transformation to select and apply at each node depends on the subtree surrounding it, and (2) even the same tree transformation may operate differently in different contexts, so it's important to recognize which specific context poses problems for the student.

We then construct a two-part Bayesian network using Bayesian Knowledge Tracing (BKT) [22] similar to that of the ANDES tutor [4]. This architecture is summarized in Figure 1. The domain-general network encodes long-term knowledge and represents the system's assessment of the student's rule mastery after the last performed exercise. It consists of two kinds of nodes: Rule node, which conveys the student's rule mastery in general, and Context-Rule node, which conveys rule mastery in a specific context. Both have as value a mastery probability $0 \leq p \leq 1$, while the conditional probability of each Context-Rule given its parent Rule is

$$P(\text{Context}_i \mid \text{Rule} = T) = 1,$$
$$P(\text{Context}_i \mid \text{Rule} = F) = \text{diff}_i,$$

where diff_i is the difficulty of context i, determined by the number of errors in context i divided by the total number of time that such context occurs (in the pre-test).

The task-specific network encodes the student's rule mastery in a specific exercise. We employ three kinds of nodes: Context-Rule, Fact and Rule-Application. Each Fact node expresses a property of the current tree, i.e., the current node is black or the parent node is red. These nodes represent the hypotheses that the student is aware of what to look for in the preconditions of the next step. The Rule-Application node has a boolean value, which is set to True if the student applies the rule correctly, and False otherwise. In essence, the system analyzes the current context, expressed by the Fact nodes, to bring up the corresponding Context-Rule node, whose probability value is used to predict the student answer's correctness. After the student submits the answer, the system records whether the prediction is right or wrong and updates the posterior value of the Context-Rule node, according to BKT. The rationale is that if the student previously made an error in a particular context, when that context shows up again in the current exercise, we would like to see whether the same error occurs. If no error is made, the student's mastery in this context

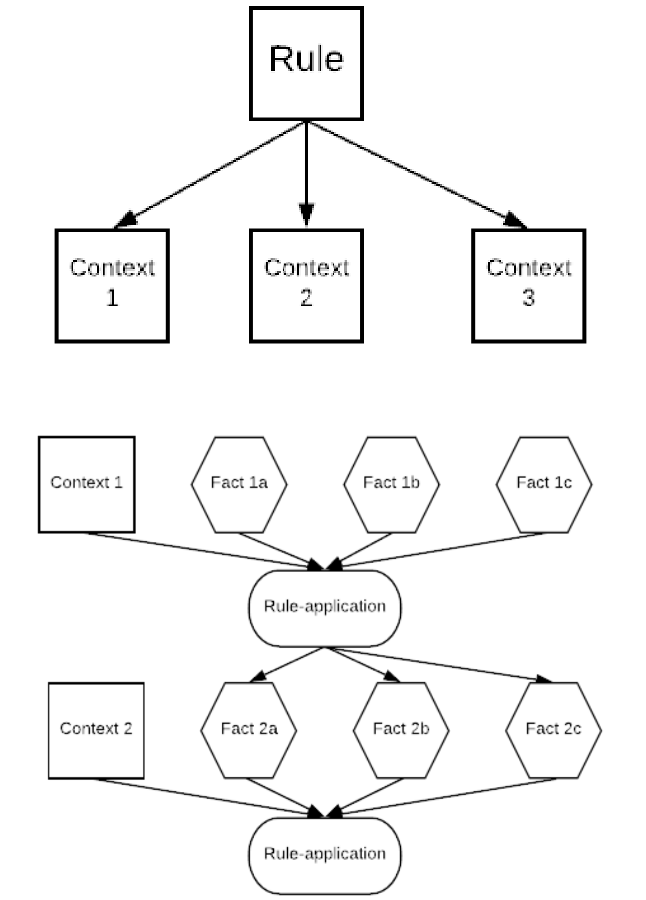


Figure 1: The structure of the domain-general network (top) and task-specific network (bottom).

has improved. This mechanism is expressed by the weight of the edge leading to each Rule-Application node:

$$P(\text{Rule-Application} = T \mid \text{all parents} = T) = 1 - P(S),$$
$$P(\text{Rule-Application} = T \mid \text{at least one parent} = F) = P(G),$$

where $P(S)$ and $P(G)$ are the slip and guess probabilities, which are part of the BKT parameters and set to a default value of 20%.

Once the student finishes an exercise, its task-specific network is discarded, but the context rule mastery probabilities are saved back to the domain-general network, so that they can be used as prior probabilities for future exercises.

5.3 Bayesian student model with extended error contexts

So far we have considered each transformation in isolation, but the nature of the solution to a red-black tree problem is a sequence of transformations, one following another. Our third approach experiments with the idea that the correctness of a student's answer may also depend on her previous answer. We perform pre-test analysis and Bayesian modeling as described in Section 5.2, but now the error context includes both the surrounding subtree and the previous transformation. With this distinction, there will be more contexts to analyze, and we would like to see how it affects the sytem's accuracy.

6. EVALUATION & RESULTS

We evaluated our approaches on four semesters of data from students in a computer science class at our institution. The semester enrollments are 20 (Fall 2016), 50 (Spring 2017), 26 (Fall 2017) and 33 (Spring 2018).

The pre and post tests are identical in content, both consisting of a small number of exercises in which students attempt to insert (delete) a node, given a starting tree. Problems in the insertion tutor require students to insert 9 numbers to an empty tree. Similarly, problems in the deletion tutor require students to delete all values from an initial tree with 9 nodes. The number of questions in each session is listed in Table 1.

	Pre-test	Tutor	Post-test
Insertion	4	20	4
Deletion	7	25 (F2017, S2018) 20 (others)	7

Table 1: Number of questions in each session. Each question has 9 parts, each of which requires multiple steps to solve.

In the tutoring section, Each time the student submits an answer, the system attempts to predict whether that answer is correct, based on the student model's knowledge. Then the actual grading is performed to check whether this prediction is right. The accuracy of the student model is defined as the number of correct predictions divided by the total number of predictions. In all subsequent tables, unless otherwise specified, the data are averaged across all students in each semester.

6.1 Evaluating performance prediction accuracy

6.1.1 Baseline prediction

When predicting with fixed probability, the resulting average accuracies approximate 50% in all semesters, with small standard deviations (5%).

6.1.2 Bayesian model with error contexts

We evaluate the Bayesian student model on both the insertion tutor and deletion tutor (Table 2). The columns, from top to bottom, respectively refer to the followings: number of average and total correct predictions, mean accuracy, standard deviation of accuracy, lowest and highest accuracy across all students in the semester.

Note that because we decided to add five more exercises in Fall 2017 and Spring 2018, the number of answers submitted (and the number of predictions) in this semester is higher than in the others. We can see that data across the fall semesters are consistent. There is more variation in Spring 2017 due to the larger number of students enrolled, but only

Insertion	F2016	S2017	F2017	S2018
Correct/Total	268/372	259/399	267/371	272/375
Accuracy	72%	66%	72%	73%
Stdev Acc	4%	8%	5%	5%
Min Acc	63%	50%	62%	65%
Max Acc	81%	86%	83%	84%

Deletion	F2016	S2017	F2017	S2018
Correct/Total	270/383	268/383	351/461	360/472
Accuracy	70%	70%	76%	74%
Stdev Acc	5%	4%	4%	4%
Min Acc	64%	61%	68%	65%
Max Acc	82%	80%	83%	81%

Table 2: System's accuracy on the insertion tutor and deletion tutor, using Bayesian modeling.

in the insertion tutor. The system achieves the highest accuracy (76%) when predicting performance in the deletion tutor of Fall 2017 - this can be explained by the increased number of exercises, which allows the Bayesian network more opportunities to update itself and to yield better predictions in turn. Overall, using Bayesian modeling yields a 20% improvement in accuracy, compared to baseline prediction.

6.1.3 Bayesian model with extended error contexts

Table 3 shows the model's accuracy when accounting for the previous transformations in the contexts. The average accuracy is around 80%, while the maximum accuracy can reach as high as 96% (in Fall 2017). Hence this approach has by far yielded the best accuracy, about 10% more than using Bayesian model with the standard error context, and 30% more than baseline prediction.

Insertion	F2016	S2017	F2017	S2018
Correct/Total	310/372	325/399	322/371	330/375
Accuracy	85%	81%	87%	83%
Stdev Acc	7%	7%	7%	8%
Min Acc	63%	62%	58%	60%
Max Acc	92%	94%	96%	92%

Deletion	F2016	S2017	F2017	S2018
Correct/Total	320/383	305/383	388/461	390/472
Accuracy	82%	79%	85%	81%
Stdev Acc	7%	8%	7%	7%
Min Acc	62%	57%	65%	61%
Max Acc	91%	87%	90%	88%

Table 3: System's accuracy on the insertion tutor and deletion tutor, using Bayesian modeling with extended error context.

We then performed additional analysis in this direction to see whether there is room for improvement and what problem-solving patterns students might have. Table 4 breaks down the accuracy in more detail; each prediction is categorized as either correct (C), false positive (FP) or false negative (FN). False positive occurs when the student answer is incorrect but predicted to be correct; false negative occurs when the student answer is correct but predicted to be incorrect. We see that in most cases, if the student is correct, the system can predict so. The majority of incorrect predictions occur in the false positive condition, where the system thinks that the student has mastered the transformation but in actuality the student still has an erroneous model. This suggests that we may be able to fine-tune the Bayesian network's behavior, in particular by decreasing the conditional probability that the student can submit a correct answer if the system thinks she understands the corresponding transformation.

Insertion	F2016	S2017	F2017	S2018
C	85%	81%	87%	85%
FP	11%	14%	10%	13%
FN	4%	5%	3%	2%

Deletion	F2016	S2017	F2017	S2018
C	82%	79%	85%	80%
FP	15%	16%	13%	14%
FN	3%	5%	2%	6%

Table 4: System's prediction results on insertion tutor and deletion tutor, averaged by students.

Next, we look at the cumulative statistics for each semester. Specifically, we would like to know the transformations involved in the answers that the system can predict accurately and in those that the system cannot. Table 5 breaks down this information from Fall 2017 based on the three categories C, FP and FN mentioned above. Here the tree insertion transformations of interest are Insert node (Insert), Color flip (Cflip), Single rotate (SingleR), Double rotate (DoubleR). Data from the other two semesters are also similar.

	Insert	Cflip	SingleR	DoubleR
C	3353 (90%)	792 (73%)	331 (79%)	348 (80%)
FP	327 (9%)	229 (21%)	59 (14%)	66 (15%)
FN	53 (1%)	71 (6%)	31 (7%)	23 (5%)
Total	3733	1092	421	437

Table 5: System's prediction result count for insertion tutor, cumulative in Fall 2017.

	Delete	Cflip	SingleR
C	3413 (89%)	786 (71%)	341 (74%)
FP	385 (10%)	243 (22%)	79 (17%)
FN	52 (1%)	78 (8%)	41 (9%)
Total	3850	1107	461

	DoubleR	DropR	Switch
C	367 (80%)	292 (68%)	795 (95%)
FP	54 (12%)	101 (24%)	27 (3%)
FN	33 (7%)	36 (8%)	15 (2%)
Total	454	429	837

Table 6: System's prediction result count for deletion tutor, cumulative in Fall 2017.

Table 6 presents the same kind of data for the deletion tutor in Fall 2017. Here the tree transformations of interest are Delete node (Delete), Color flip (Cflip), Single rotate (SingleR), Double rotate (DoubleR), Drop rotate (DropR) and Switch value (Switch).

We also look at, among all the error contexts identified, which pair of sequential transformations (i.e., the current transformation following a previous transformation) occurs the most, since our analysis includes the previous transformation in the error contexts. Table 7 shows that, in red-black tree insertion, students are most likely to make mistakes in rotation operations if they previously performed an insert node operation. This pattern can be explained by the fact that in most tree insertion problems, the final step is to insert a new node at a leaf node's child. However, in some cases, this leaf is already red; adding a red child to it would then yield two consecutive red nodes, violating the properties of red-black trees. Hence another rotation at the newly inserted node is required to remedy the situation, which students tend to forget. It should be noted that a color flip may also result in consecutive red nodes, thereby forcing a rotation to follow; the third and fourth row in Table 7 represent this case. In general, from our teaching experience, all four cases occur very often, but this is the first time we obtain a relative ranking of their frequencies.

Transformation	Previous Trans	Count
SingleR	Insert	90
DoubleR	Insert	72
SingleR	Cflip	65
DoubleR	Cflip	50

Table 7: Most common pairs of insertion transformations in students' errors across three semesters.

Table 8 shows that, in red-black tree deletion, students are most likely to make mistakes in *delete node* following *switch value*. Interestingly, as we previously analyzed, students usually perform *switch value* correctly. However, after this step, they tend to move straight to the leaf whose value was switched and delete it - this is correct in normal binary search trees, but in red-black trees, we still have to traverse down one node at a time until reaching the leaf, performing necessary transformations along the way before the actual deletion. Another noteworthy point is that students tend to forget to execute the *drop rotate* operation, but only when it is necessary to do so at the root (in this case, drop rotate is the first transformation in the solution sequence, so it has no previous transformation).

Transformation	Previous Trans	Count
Delete	Switch	98
DoubleR	Cflip	78
SingleR	Cflip	76
Drop rotate	-	27

Table 8: Most common pairs of deletion transformations in students' errors across three semesters.

6.2 Assessing students' test performances

While the previous study by Liew & Xhakaj [15] reported an improvement in individual student performance from pre-test to post-test, it was conducted on a small sample of 12 students. To measure this effect on a larger scale, we performed a paired samples t-test to compare the student's number of first errors in the pre-test e_{pre} and in the post-test e_{post}. Results show that in tree insertion, there was a significant difference between e_{pre} $(M = 2.81,\ SD = 1.35)$ and e_{post} $(M = 1.72, SD = 1.35)$; $t(137) = -8.23$, $p = 2.95 \cdot 10^{-13}$. Similarly, in deletion, there was a significant difference between e_{pre} and e_{post} $(M = 3.63,\ SD = 1.08)$ and e_{post} $(M = 2.68,\ SD = 1.48)$; $t(137) = -5.33$, $p = 3.12 \cdot 10^{-7}$. Hence the impact of the tutoring system on reducing student errors is statistically significant at the 1% level, which is consistent with [15].

Further analysis on the total number of errors overall and per each transformation rule reveals that the errors in node selection decrease across all semesters; in insertion exercises there is a steady 50% reduction from pre test to post test, whereas the differences vary more in deletion exercises. Interestingly, the number of errors in applications do not seem to decrease by much; in particular, errors in *single rotation* and *double rotation* do not decrease significantly, and even increase in some cases, between the pre and post test. The reason is that in the pre-test, because most students forget about *color flip*, they do not have many opportunities to apply *single rotation* or *double rotation*, resulting in few application errors reported. On the other hand. in the post-test, students have already mastered *color flip*, which then prompted them to apply rotations on more occasions, in which case more application errors were likely to occur. On further analysis, if we only consider students who did make rotation errors in the pre-test, then their number of rotation errors in the post-test also decreased significantly, by almost 75%. A more detailed breakdown of students' test performance is presented in [13].

7. CONCLUSION

This paper has described how we have mined logs of student actions on red-black tree operations to build a Bayesian model of their mastery of the skills involved. The analysis of the logs has helped us to determine (1) the most frequent errors that the students make, and (2) the contexts in which the errors are made. This knowledge can and will be used to improve both the tutoring system and the classroom instruction. The instructors can use the data to modify and customize their instruction to focus more attention on the problematic areas.

Results from this study also open up several future directions. First and foremost, the student model has demonstrated a reasonable performance and can now be used to build an adaptive learning system, which can potentially further reduce the number of errors. Second, gathering more student data would allow the implementation of more sophisticated techniques, such as hierarchical Bayesian learning or deep learning, in student model construction, which would in turn enhance the model's accuracy. Finally, balanced trees in general share many common properties and transformations; an adaptation of the current system to a related domain (e.g., AVL trees, AA trees, splay trees), could therefore provide insights on how general the underlying framework is.

8. REFERENCES

[1] Almond, R. G., Mislevy, R. J., Steinberg, L. S., Yan, D., and Williamson, D. M. *Bayesian networks in educational assessment.* Springer, 2015.

[2] Bauer, A., and Popović, Z. Collaborative problem solving in an open-ended scientific discovery game.

Proc. ACM Hum.-Comput. Interact. 1, CSCW (Dec. 2017), 22:1–22:21.

[3] Cheang, B., Kurnia, A., Lim, A., and Oon, W.-C. On automated grading of programming assignments in an academic institution. *Computers & Education 41*, 2 (2003), 121–131.

[4] Conati, C., Gertner, A., and Vanlehn, K. Using bayesian networks to manage uncertainty in student modeling. *User modeling and user-adapted interaction 12*, 4 (2002), 371–417.

[5] Cormen, T. H. *Introduction to algorithms*. MIT press, 2009.

[6] Forlizzi, J., McLaren, B. M., Ganoe, C., McLaren, P. B., Kihumba, G., and Lister, K. Decimal point: Designing and developing an educational game to teach decimals to middle school students. In *8th European Conference on Games-Based Learning: ECGBL2014* (2014), pp. 128–135.

[7] G H Al-Bastami, B., and Abu Naser, S. Design and development of an intelligent tutoring system for c# language. 8795–8809.

[8] Geigle, C., Zhai, C., and Ferguson, D. C. An exploration of automated grading of complex assignments. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale* (2016), ACM, pp. 351–360.

[9] Helmick, M. T. Interface-based programming assignments and automatic grading of java programs. In *ACM SIGCSE Bulletin* (2007), vol. 39, ACM, pp. 63–67.

[10] Kastner, M., and Stangla, B. Multiple choice and constructed response tests: Do test format and scoring matter? *Procedia-Social and Behavioral Sciences 12* (2011), 263–273.

[11] Kyrilov, A., and Noelle, D. C. Using case-based reasoning to improve the quality of feedback provided by automated grading systems. *International Association for Development of the Information Society* (2014).

[12] Lee, Y.-J. Analyzing log files to predict students' problem solving performance in a computer-based physics tutor. *Journal of Educational Technology & Society 18*, 2 (2015).

[13] Liew, C. W., and Nguyen, H. Determining what the student understands - assessment in an unscaffolded environment. In *Proceedings of the Fourteenth International Conference on Intelligent Tutoring Systems (ITS)* (2018).

[14] Liew, C. W., and Smith, D. E. Checking for dimensional correctness in physics equations. In *FLAIRS Conference* (2002), pp. 299–303.

[15] Liew, C. W., and Xhakaj, F. Teaching a complex process: Insertion in red black trees. In *International Conference on Artificial Intelligence in Education* (2015), Springer, pp. 698–701.

[16] Martin, J., and VanLehn, K. Student assessment using bayesian nets. *International Journal of Human-Computer Studies 42*, 6 (1995), 575–591.

[17] Matthews, K., Janicki, T., He, L., and Patterson, L. Implementation of an automated grading system with an adaptive learning component to affect student feedback and response time. *Journal of Information Systems Education 23*, 1 (2012), 71.

[18] Minaei-Bidgoli, B., Kashy, D. A., Kortemeyer, G., and Punch, W. F. Predicting student performance: an application of data mining methods with an educational web-based system. In *Frontiers in education, 2003. FIE 2003 33rd annual* (2003), vol. 1, IEEE, pp. T2A–13.

[19] Mitrovic, A., Ohlsson, S., and Barrow, D. K. The effect of positive feedback in a constraint-based intelligent tutoring system. *Computers & Education 60*, 1 (2013), 264–272.

[20] Piech, C., Huang, J., Nguyen, A., Phulsuksombati, M., Sahami, M., and Guibas, L. Learning program embeddings to propagate feedback on student code. *arXiv preprint arXiv:1505.05969* (2015).

[21] Singh, R., Gulwani, S., and Solar-Lezama, A. Automated feedback generation for introductory programming assignments. *ACM SIGPLAN Notices 48*, 6 (2013), 15–26.

[22] Yudelson, M. V., Koedinger, K. R., and Gordon, G. J. Individualized bayesian knowledge tracing models. In *International Conference on Artificial Intelligence in Education* (2013), Springer, pp. 171–180.

Your Actions or Your Associates? Predicting Certification and Dropout in MOOCs with Behavioral and Social Features

Niki Gitinabard, Farzaneh Khoshnevisan, & Collin F. Lynch
North Carolina State University
Raleigh, NC, USA
{ngitina, fkhoshn, cflynch}@ncsu.edu

Elle Yuan Wang
Columbia University
New York City, NY, USA
elle.wang@columbia.edu

ABSTRACT

The high level of attrition and low rate of certification in Massive Open Online Courses (MOOCs) has prompted a great deal of research. Prior researchers have focused on predicting dropout based upon behavioral features such as student confusion, clickstream patterns, and social interactions. However, few studies have focused on combining student logs with forum data. In this work, we use data from two different offerings of the same MOOC. We conduct a survival analysis to identify likely dropouts. We then examine two classes of features, social and behavioral, and apply a combination of modeling and feature-selection methods to identify the most relevant features to predict both dropout and certification. We examine the utility of three different model types and we consider the impact of different definitions of dropout on the predictors. Finally we assess the reliability of the models over time by evaluating whether or not models from week 1 can predict dropout in week 2, and so on. The outcomes of this study will help instructors identify students likely to fail or dropout as soon as the first two weeks and provide them with more support.

1. INTRODUCTION

Massive Open Online Courses (MOOCs) can provide broad and potentially scalable platforms for learning. Truly open MOOCs allow students around the world to enroll in any course that piques their interest or meets professional needs. Most of the available MOOCs are free, and many stay open perpetually even after their official offerings are complete thus allowing students to use them as a regular reference point or as a social platform.

One major concern with MOOCs is that they have extremely high rates of dropout. More than 85% of students who register for a MOOC quit without completing it [17]. Prior research has indicated that student dropout in MOOCs, and student performance more generally, is highly correlated with features of the students' online activities such as viewing lectures or attempting mastery quizzes [22, 3, 16, 5, 20, 24, 23, 28, 29, 13, 9, 1, 6, 8, 1, 14, 15]. These activities can be classified as student-system interactions (e.g. video viewing) [22, 3, 20] and student-student interactions (e.g. posting to a forum) [16, 5, 23, 8, 1, 14, 15].

Social network analyses of interactions among students has shown that students' social interactions and social presence metrics can be used to predict their performance [16, 5, 23, 30]. However, in most of these studies, the authors did not focus on how the students form their social networks over time. Nor did they examine whether or not the different types of user forums produced substantively different networks. Similarly, prior studies of dropout prediction from activity logs have shown that students' study habits can be used to predict attrition [9, 28, 28, 1, 24]. However activity logs and social metrics cover very different aspects of student behavior. Therefore it is possible that by combining the two, we may be able to improve our insights into students' behaviors and thus, improve our ability to predict both performance and dropout. Few researchers have combined behavioral and social metrics to improve prediction performance [9, 25]. Thus it is beneficial to make this comparison on new datasets to check the generality of the outcomes.

Prior researchers have also shown that students' actions during the first few weeks of a course can be used to predict their subsequent performance [3, 9, 20, 25, 15]. It has also been shown that models trained on one class can sometimes be applied to other classes [3, 26, 2], but these findings have only been tested on a few MOOCs and are not yet reliable. Therefore it is an open question whether metrics of the type that we consider will be transferable.

In this study, we used two different offerings of a MOOC on Big Data in Education, offered by Dr. Ryan Baker on the Coursera Platform in 2013 and EdX in 2015. We generated social networks based upon two approaches taken by the prior studies for the same dataset based on different sets of assumptions, compared them and show how changing assumptions can affect the findings and also, how forum structure can help us make assumptions with more caution [30, 5]. We also perform a survival analysis to find the groups of students that are more likely to dropout and compare the findings among both classes. Then we use feature selection to find out which features can provide us with more information gain. Later, we train predictive models using the top features in our feature selection and predict dropout and certification. Finally, we use the prediction models trained on each week of the first offering of the earlier course to predict dropout and certificate earning early in the second offering.

Overall, we aim to investigate the following research questions:

1. What features are most predictive of student drop-out?
2. How will the choice of target label, social graph generation, and features affect prediction results?
3. How early can we predict student dropout in MOOCs?
4. Can we make predictions across course offerings by using models trained on one year to predict others?

As part of this work we will show how important the assumptions we make are on the performance and findings of the study. The generated models can also help MOOC instructors identify students who are likely to dropout early in the semester using models from prior classes and provide the students with more support and motivation to complete the course.

2. BACKGROUND

Prior research on 39 MOOCs showed that on average only 6.5% of the users who enroll in a MOOC finish with a passing grade and earn a certificate [17]. As Yang et al. noted, this high attrition may be caused by several factors such as students losing interest over time, or by mounting confusion and frustration. Or it may simply be the case that they never intended to complete the course in the first place [28]. We acknowledge that some users enroll in MOOCs only to access specific parts of the material and with no intention of obtaining a certificate and that intention to finish the course is correlated with course completion [22, 12]. Pursel et al., for example, showed that students' plans to watch videos and earn a certificate is a significant predictor of their course completion [22]. Gutl et al. surveyed students who did not complete a course and found that only 22% of them had intended to do so in the first place [12].

In addition to intentions and motivation, researchers have also observed that other attributes are useful for predicting students' course completion. These include: the number of videos that a student watches in a week; the number of quiz or assignments they attempt; the number of forum posts made per week along with the post length; the time spent on assignments; whether they spend more time on forums or on the assignments; whether or not they start early; and demographic data such as their age, fluency with English, and their education level [22, 9, 23, 29, 1, 6, 24]. Some researchers have also utilized social network metrics such as degree, centrality, hub, and authority scores [11, 14, 29, 16, 5, 23, 30].

Joksimovic et al. showed that students' social presence metrics can be used to predict their final grades [16]. Some examples of these parameters include: continuing a thread, complimenting other users, and expressing appreciation. Eckles et al. went further than general graph attributes and observed that whether or not a students' best friend stays in the course is strongly correlated with whether or not they do so [8]. Unlike other researchers, Eckles et al. did not use a social network to define this relationship but surveyed the students directly. Brown et al. analyzed the same 2013 dataset that we use here. They showed that students form communities based on their interactions on the discussion forum and membership of these groups are correlated with the students' grades [4]. In the prior literature, different methods have been used to generate social networks, but few comparative studies have been done to highlight their effects. Brown et al. [4] and Zhu et al. [30] exemplify some of the alternatives. Brown et al. formed a weighted undirected social network by connecting each author that posts to a discussion thread with all of the authors that had previously contributed to it, on the assumption that each author reads the current thread before adding to the conversation and that the reply is intended for all authors [5]. Thus, the graph assumes an implicit social connection by virtue of the group conversation. Zhu et al., by contrast, added a connection from each author who contributes to a thread to the author of the first post alone on the assumption that the thread consists of a series of flat replies to the original post and that users will only read the first post before replying [30]. Whether or not these assumptions are valid depends upon the structure of the forums and the habits of the students themselves. Indeed they depend upon the "culture" of the class. It is therefore important to study the impact of these assumptions on the outcome of a study.

Prior research has shown that these predictive models can not only be used to predict students' performance based upon the data from the entire semester in the same class, that they can also be used to make early predictions, based upon partial class data, or to make predictions across classes. Previous studies used a model trained on one offering of a MOOC to make predictions for another [2, 3]. An early notifier to identify student performance in the course using only the first few weeks of data in MOOCs has also been investigated before [3, 15].

As prior research shows, both behavioral and social features are predictive of dropout. These features cover different aspects of student activities, we therefore decided to use a selection of both types of features to train our predictive models. Fei et al. used a combination of these features to generate predictive models, but they did not evaluate this hybrid approach against pure activity or social models [9]. Taylor et al., however, has shown that in their MOOC, the addition of forum activity did not add much value to a previous log-based predictive model [25]. It is therefore important to study whether or not combining these feature types can make a difference in different courses because it might depend on the course structure and its use of the discussion forum.

3. DATASET

We analyzed data from two different offerings of the "Big Data in Education" MOOC (BDE MOOC), from 2013 and 2015. Table 1 presents some basic characteristics of these two datasets. The presentation and storage formats were slightly different as in 2013 it was offered on the Coursera platform while in 2015 it was deployed on EdX. We will therefore describe them separately in the following sub-sections.

"Big Data in Education" course was offered by the Teacher's College at Columbia University on the Coursera platform in 2013. A total of 55,013 students registered for the course, but only 17,295 had any activity recorded in the logs. Only 750 students made one or more posts or replies on the discussion forum. Our dataset does not include view records so we cannot estimate how many students visited the forum but made no contribution. Roughly 1,599 students submitted assignments or quizzes. In this study, we considered 23,080 students who had at least a recorded activity in the forum, assignment submission, or lecture view. Both of the courses were open for students after the official offering was over. Therefore the datasets included students who worked on their own well after the instructor and the rest of the class had left. For this analysis we decided to focus solely on those students who started and finished the courses on schedule so that their activities would fit properly into the official weeks and the course calendar. This left a total of 17,295 students remaining in our dataset. We extracted the grades for these remaining students. Among all, only 1,381 had non-zero final grades.

This class was offered again on the EdX platform in 2015. As before, the provided data consisted of activity logs, final certificates, and forum posts. In addition to the threaded discussion forum, edX also offers a chat platform among participants of the course called Bazaar where a lot of the discussions among students take place. Unfortunately, the data from that platform was not available for this study. A total of 10,190 students were initially enrolled in this class. Only 519 students posted or replied on the forum, 1,437 submitted at least one of the problems, and 320 students had a non-zero final grade. As with the 2013 dataset, we removed the students who had submissions before or after the course dates leaving 5,077 students.

Data	Enrolled Students	Forum Active Students	Students Who Had Some Submissions	Number of Forum Posts	Students Who Had Some Activity	Non-zero Grades	Earned Certificates	Thread Count	Thread Avg Length	Thread Max Length	Thread Min Length
BDE 2013	55,013	750	1,599	4,261	17,295	1,381	638	281	5.31	89	1
BDE 2015	10,190	519	1,437	2,063	5,077	320	117	624	2.24	36	1

Table 1: BDE MOOC 2013 and 2015 Characteristics, Including Only the Students Who Started and Finished the Course On-schedule

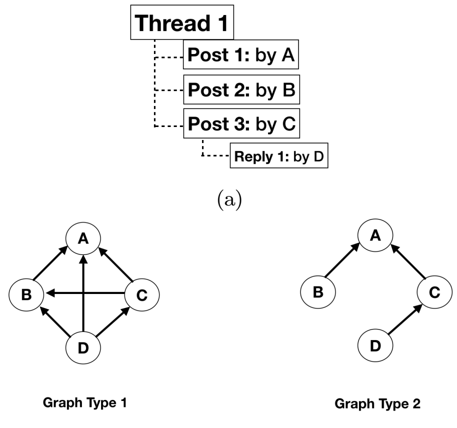
4. METHODS

We began our analysis by generating a social network, and extracting structural and behavioral features from it and the logs. We ran feature selection to determine whether or not a combination of these features can improve the performance of the overall model, when compared to using each of the groups separately. In the final step of this process we rah a machine learning analysis to predict dropout and certification. We extracted each of the features on a week-by-week basis. Thus we produced a set of per-week datasets each of which includes all data before the end of the associated week. This will help us to analyze how early we can predict dropout and certificate earning based on their activities so far. We will discuss each of these steps in the following subsections.

4.1 Graph Generation

In both classes the forums consist of a series of threaded discussions. Class participants may initiate a thread by making a root post and may reply to existing threads by adding comments at the end or by replying to a specific post. As mentioned above, two approaches have been used to generate social graphs from discussion forums. Brown et al. connected authors of all the posts and replies in a thread to authors of all the preceding contributions in the thread [5]. This method assumes that everyone who posts on a thread or replies to a post has read all of the preceding posts on the same thread first and is responding to all of them. Another approach used by Zhu et al. suggests connecting all the authors in a thread to the author who originated it [30]. This approach is more reasonable for flat forums where each thread is a separate question and all of the replies are directed towards the first post. In this study, we generated the social graphs based upon both approaches. We designate Brown's approach "Type 1" and Zhu's approach "Type 2". Figure 1 shows an example of a thread structure and the two corresponding graphs to highlight differences between these methods. We then compared these graphs in terms of their ability to predict both dropout and whether or not students would earn a certificate, only among those who lie on the graph. The structures of the forums differ between Edx and Coursera, so we expect that this difference will be reflected in the relative performance of the classifiers on these graphs. On the Coursera platform, used for BDE 2013, clicking on the first post in a thread will show all of the remaining posts as well as replies to them. Thus it makes sense to construct a Type 1 graph and to connect every author to the authors of the preceding posts. However, the structure of the EdX forum is slightly different. Once a thread is selected, you see the beginning of all the posts but not the full text. By selecting each post you can view the full content and the replies. Therefore, when reaching a specific post, the users do not necessarily need to view preceding comments. In this case, it seems more reasonable to construct a Type 2 graph by connecting replies to the original post alone.

The length and the number of threads for each class is shown in Table 1. In 2013 there were fewer threads than in 2015 but the threads themselves were generally longer. This may be a consequence of the difference in the platforms, the nature of the discussion forums, the addition of the chat platform, or how the users learned to interact with the tools.



(b) (c)

Figure 1: Graph construction of Type 1 and 2 for post/reply structure example

Since we are focused on student to student interactions we chose to remove the instructor from the graphs. We also removed all of the isolated nodes (students who did not make posts or receive replies) before calculating the social metrics as all metrics for an isolated node would be zero.

4.2 Generated Features

For each student in the graphs we calculated the following features: *Betweenness Centrality* showing to what extent a vertex lies on the paths between other users [10], which indicates the importance of the student in connecting other students together; *Hub score* showing the extent that a node points to many good authorities [19], students with higher hub scores, respond to active students' posts more frequently; *Authority score* showing the extent that a node is pointed by many good hubs [19], students with higher authority scores, receive comments from hub students more frequently; *In-degree* showing the number of connections a student has received by getting replies from others; *Out-degree* showing the number of connections the student has made by posting replies to others; and *Dropped_out_neighbors* showing the proportion of a user's neighbors that have already dropped out in each week. This metric was inspired by Eckles et al. [8], and was defined as a way to estimate whether or not the students' attrition can be affected by their closest neighbors. This feature can show how much a user has been exposed to unmotivated users.

In addition to the social features described above, we defined other general features based upon students' log data and forum activity. Some of these, which we call *forum features*, are based on activities on the forum including the *total_posts*, *total_comments*, as well as the total number of *votes* (total upvotes − total downvotes) that the student received on their posts. The third group of features, called *behavioral features*, is extracted from the activity logs. We extracted the total *video_views* and *video_downloads* for the 2013 students class. The 2015 offering did not provide download information. In 2015 students were offered 'chapters' to view. We therefore extracted the total number of *video_views* and *chapter_views* for this class. The *total_attempts* is also included in both cases. This represents the total number of assignment submissions for each student.

The last group of extracted features, which is our target for prediction, includes *semester_dropout*, *week_dropout*, *inactive_next_week*, and *certificate*. Defining dropout based on observations of online activities is not trivial because the students do not explicitly declare their leaving. Prior studies have proposed different measures reflecting dropout [9, 28]. We define our measures similar to Fei et al. as described below and generate our predictive models based on all of them [9]. Mostly our focus in this paper will be on semester dropout and certificate earning because they provide a static label for students over all the weeks of the semester.

Semester dropout: Will this student stop engaging at some point? This feature is represented by a boolean flag which indicates that the student dropped out of the course *before the end*. Thus if a student quits performing actions in the course in any week but the last then this will be set to 1 for all weeks. We do not consider students with no activity in the last week as dropout since they may have finished earlier in the week.

Week dropout: Will this student stop working from next week? This is a boolean flag that is used to designate when a student drops out. It will be set to 1 for a week if the student does not perform any activities in the subsequent weeks. The activities we consider include: posting or commenting on the forum, submitting assignment, and watching or downloading lecture videos (or chapter view in BDE 2015 data).

Inactive next week: shows whether the student will be inactive in the following week.

Certificate: shows whether the student has earned a certificate.

4.3 Survival Analysis

Survival analysis is the analysis of data involving the time remaining to the occurrence of some event of interest. This method was originally introduced in medical research and is used to predict how long patients would survive, or go without some change, based upon their data [21]. It has since been adapted to a number of other fields where estimating the time until the occurrence of an event or a boolean flag is of interest [28]. One objective of survival analysis is to examine whether the survival times are related to other features. For this purpose, regression models can be used to assess the effect of covariates on an outcome. In this study we used a multivariate version of Cox proportional hazards model to fit the hazard ratio at time t as follows:

$$h(t;x_1,...,x_n) = h_0(t)e^{\beta_1 x_1 + ... + \beta_n x_n} \tag{1}$$

In this formula, $h_0(t)$ is the baseline hazard that is the hazard ratio of an individual, at time t where all the covariates are zero. The effect of variable x_k while all other variables are fixed is interpreted as: for each unit increase in x_k with all other variables held fixed, the hazard is multiplied by e^{β_k} [18].

Here, we used the week of dropout as the target time. For the students who have not dropped out until week 6, we consider them as not dropping out, via right censoring. *Right censoring* occurs when an individual has not had the event of interest until the end of study. Further, we normalized all the variables to have a mean of zero and a standard deviation of one. Therefore, the resulting hazard ratio of 1.2 for total number of attempts for example, shows that students with one standard deviation higher attempts than average are 20% more likely to dropout. Likewise, a hazard ratio of 80% would show the users are 20% less likely to dropout. For this study, we generated two models, the first included social features alone while the second one included all of the predefined features.

4.4 Dropout Prediction

As mentioned earlier, we generated multiple datasets for each class, each of which corresponds to the activities up to the end of a given week, starting from the first week of the course. We then built classifiers for each dataset to evaluate their performance. Our primary goal was to determine how early the data would be sufficient for the model to train and to predict the outcome with high accuracy. In both classes, the percentage of students who stayed engaged in the course until the end was less than 20% and the ones who earned a certificate were around 2%, so a model trained on that data would generate biased results. To make the dataset balanced, we randomly removed some of the majority class instances until the number of users in both the classes were equal. For the prediction, we began by employing decision tree-based feature selection based on Gini impurity index to select the most important features for the prediction of each target class [7]. We then applied Support Vector Machine (SVMs) and Logistic Regression (LR) for the classification task. Since Logistic Regression outperformed SVM in all cases, we only report the AUC performance of the logistic regression when comparing different models. In order to tune the parameters and validate the training procedure, we applied 10-fold nested cross-validation to estimate each outcome.

5. RESULTS AND DISCUSSION

5.1 Graph Construction

The construction method used when making a social graph can affect the predictive performance. In this section, we assess the predictive power of the graph features, for both dropout and certification, that we extracted from the two different social graph types. Table 2 presents the predictive performance of an LR classifier trained on the aforementioned features extracted from the two graph types for BDE 2013 class. As we hypothesized, for both semester dropout and certificate prediction, *Type 1* graph features perform slightly better than the *Type 2*.

Class	Target	AUC		F-measure	
		Graph 1	Graph 2	Graph 1	Graph 2
BDE 2013	Semester Dropout	0.72	0.707	0.709	0.689
	Certificate	0.808	0.796	0.763	0.749
BDE 2015	Semester Dropout	0.548	0.577	0.559	0.666
	Certificate	0.545	0.607	0.491	0.676

Table 2: Graph Construction Effect on Dropout and Certification

Table 2 shows the same comparisons for the BDE 2015 class offered on the EdX platform. Consistent with our hypothesis, the predictive model based on graph 2 features performs considerably better than the graph 1-based model. Thus, we will use this construction of the graph for dropout and certificate prediction later. Overall, the graph features in BDE 2015 are less predictive of student outcomes than with the BDE 2013 data. Similar to the difference in the length of the threads, this may also be due in part to the presence of the separate chat platform where part of the discussion among students takes place. Those interactions are not represented in our dataset. As our results show, the methods and assumptions used for the generation of social graphs should be tailored to the class and forum structure and some methods may not generalize to all of the other classes or platforms.

5.2 Survival Analysis

We explored two different sets of features in our survival analysis to discover the impact of the features on dropout in the two course offerings. In BDE 2013, when including all the aforementioned features in Section 4.2, we observed that both the behavioral and social features have a high hazard ratio and significant p-values as shown in Table 3. Accordingly, the hazard ratio (HR) 0.71 for video download indicates that students who download one standard deviation (SD) more videos than average are 29% less likely to dropout compared to the ones with an average number of video downloads. Betweenness with a hazard ratio of 1.74 illustrates that the students with one SD more betweenness than average are 74% more likely to dropout. We examined some sample posts made by the students with high betweenness. It appears that many of the posts are social niceties such as expressions of gratitude or appreciation for the instructor or fellow students rather than being substantive contributions to the discussion (e.g. "Nice work" or "Your kind of persistence will always pay off eventually").

In our social model, we only considered the features that were extracted from the social graph in order to assess their effect on students' survival in the course. As our results suggest, the students whose out-degree or in-degree are one SD higher than the average are 22% and 40% less likely to dropout respectively. This means that the students who typically answer others' questions or post new questions are more likely to stay active in the course. When comparing this finding with betweenness from the previous model, we can conclude that the students with only high in-degree might be more confused, while the students with high out-degree probably understand the material better, or think that they do, and are more willing to share information. Doing both at the same time however, may show that the student is interested in socializing rather than information exchange, which may not help them to understand the material, complete the course, or gain a certificate because the socialization may take priority over learning.

Features			No grade		Social	
	Mean	SD	HR	SE	HR	SE
video_download	14.46	68.25	0.71***	0.03	—	
total_attempts	0.66	3.27	0.57***	0.04	—	
total_posts	0.03	0.34	0.63***	0.13	—	
indegree	0.27	2.67	0.75**	0.09	0.60***	0.10
outdegree	0.27	2.68			0.78*	0.09
betweenness	17.58	326.42	1.74***	0.14		

Table 3: BDE MOOC 2013 - Survival Analysis for Different Models (*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, —: not included)

The survival analysis results for the BDE 2015 course is shown in Table 4. The strongest features in this offering are largely behavioral features such as chapter views, total posts, and the total number of attempts. Chapter views had a hazard ratio of 0.53. Thus students with 1.5 more views than 2.32, are 47% more likely to continue in the course. In this case, the social features are not significant unlike the 2013 class. Additionally, having more posts in the 2013 class seemed to help people complete more, while in this class it had a negative influence. Comparing the instructor and TA activity in both classes shows that the instructor and the most active TA made many more comments in 2013 than in 2015. In 2013, the instructor and the most active TA made a total of 432 comments, while in 2015 only 133 comments were made by the instructor, and we identified no TA with significant activity. If we assume that most of the posts were expressions of confusion, the more replies that they received, the more likely it is for their confusion to get resolved. Based on the observed reply behavior of the teaching staff in those classes, it seems likely that confused students had a better chance of finding an answer in the 2013 class than in 2015. It is also possible that part of the support was provided to students via the separate chat platform, but in either case it seems that posting on the forum was less helpful in 2015 than 2013. This may indicate that posts and replies did not resolve confusion. Additionally, the results of the survival analysis align with the results of the comparison among predictive models presented in Table 2.

5.3 Feature Selection

The five most important features for each prediction task and their importance scores for BDE MOOC 2013 is shown in Table 5. As the dropout feature selection results show, video download, video view, and total attempts are the most important features for prediction of the semester dropout, while total posts and indegree are significantly less important and the remainder of the features do not show up. Therefore, we used the top three features to train our semester dropout classifier. Furthermore, when predicting certification, total attempt and video view features had the highest importance score and we used them for training the model. Similarly, in 2015, video view, chapter view, and total attempts had the highest importance score for both dropout and certificate prediction.

Features			No grade		Social	
	Mean	SD	HR	SE	HR	SE
chapter_view	2.32	1.57	0.53***	0.03	—	
total_posts	0.2	1.31	1.43*	0.14	—	
total_attempts	1.36	3.27	0.88**	0.04	—	
outdegree	0.02	0.26			0.43***	0.18

Table 4: BDE MOOC 2015 - Survival Analysis for Different Models (*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, —: not included)

	Semester Dropout		Certificate	
Rank	Feature	Importance	Feature	Importance
1	video_download	0.604	total_attempts	0.692
2	video_view	0.230	video_view	0.178
3	total_attempts	0.111	votes	0.045
4	total_posts	0.013	indegree	0.038
5	indegree	0.011	total_posts	0.019

Table 5: BDE MOOC 2013 - Feature selection using Decision Tree

Our observations showed that none of the forum features representing participation were as informative as the behavioral features. This was due in part to the fact that there was a small proportion of students who had any forum activity. Additionally, we have access to a survey completed by students before starting the course. A total of 155 and 229 students from the 682 and 483 forum active ones participated in the survey respectively in the 2013 and 2015 classes. An analysis on the responses of the forum active students shows that more than 66% of them indicated the reason for taking the class as it being relevant to their field of study, more than 77% of them indicated that it is relevant to their career, more than 87% of them believed that it will help them expand their knowledge of the field, and only less than 40% mentioned that it will help their resume. So, it seems like not many of them were concerned about finishing the course, getting a certificate, and using it as a boost to their resume. More information on the structure of the survey is available in Wang et al. [27]. Also, some more analysis on the student replies to the survey and their certificate earning is available in Andres et al. [1].

5.4 Model Performance

To train our models, we only considered features with more than a 0.1 importance score in feature selection and applied logistic regression with 10-fold cross-validation to evaluate the model. Figure 2 presents the AUC performance of each classifier over the first six weeks of the course. F-Measure performance also had a similar trend. As we observed, the certificate prediction model had an AUC above 90% from the first week of the course. While the model for semester dropout obtained an AUC of approximately 79% in the first week and gradually increased thereafter. The Week dropout and inactive next week models behaved similarly.

The classification performance of the models for the BDE 2015 dataset was similar to 2013. As with the BDE 2013 dataset, the certificate prediction performance is above 85% from the first week and improves gradually thereafter. The three dropout definitions

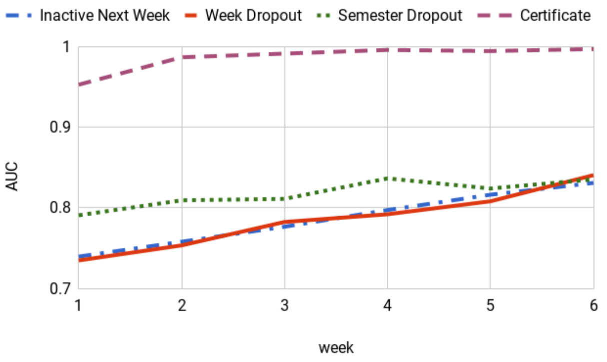


Figure 2: BDE 2013 - AUC Prediction Performance of Three Dropout Targets and Certificate

have almost the same trend and behavior while obtaining an F-measure performance around 70%.

Our results show that even though most of the more complicated metrics were removed during the feature selection method, the trained models for both of the classes were able to predict dropout with an F-Measure and AUC of ≈70% as early as the third week of the class. They were also able to predict certification with an F-Measure and AUC of ≈90% from the very first week. It also appears that the students who earn a certificate have relatively distinct behavior from that of their peers starting at the beginning of the semester.

It is interesting to note that some of the features showed significant outcomes in the survival analysis, but were not selected by the feature selection algorithm. This was surprising as one argument that has been advanced for survival analysis is that it is better at handling time-aware events. In order to evaluate this apparent conflict we trained a separate model based on the features that were significant for the survival analysis. The resulting model did not outperform the models that relied on the tree-based approach. In most cases the resulting models were comparable or the survival model underperformed. While this does not prove that survival-based selection is unusable it does merit further study.

5.5 Cross-Class Dropout and Certification

In order to evaluate the generalizability of the classification models, we took a cross-class approach. To do so, we used all of the data from each week of the first offering, based on the behavioral features common among both classes, which includes only video view and total attempts. Then we tested the model on all data from the corresponding week of BDE 2015. Table 6 presents the F-measure and AUC performance of this model over six weeks of the course for the task of semester dropout and certificate prediction. As the results suggest, the predictive power of this model is relatively high when using only the two aforementioned features, especially for the certificate prediction task. This finding suggests that even though there are some differences among these classes and the features selected for the classification of each might be slightly different, the models are able to predict students' outcome as early as the first one or two weeks with reasonable accuracy. However, these results need to be validated on other courses with multiple offerings.

	Semester Dropout		Certificate	
Week	F-measure	AUC	F-measure	AUC
1	0.606	0.764	0.727	0.885
2	0.723	0.883	0.776	0.935
3	0.706	0.882	0.741	0.879
4	0.627	0.879	0.721	0.871
5	0.490	0.915	0.750	0.861
6	0.586	0.915	0.836	0.951

Table 6: AUC and F-measure performance of Cross-class dropout and certificate prediction model

6. CONCLUSION

Our primary goal in this study was to predict student performance and dropout based upon different social and behavioral features. One focus of our work here was on the testing assumptions that are usually made when generating social features and choosing the analysis method. These findings suggest that even for similar classes with the same instructor, a change in platform or instructor/TA behavior can change the impact or appearance of student engagement in the forums. As a result, the choice of model features and the feature generation methods matter a great deal. For example, we tried two different social graph generation methods that both were suggested in the literature and based on the forum structure the better choice for each class was different. Additionally, as our results suggest, behavioral features such as submissions and video watching are better predictors of student dropout and certification than social behavior. Adding social metrics to the trained behavioral models does not seem to improve their performance because very few users seem to place any value on those features. Additionally, we observed that a behavior-based predictive model trained on a former offering is applicable to a new offering, despite the differences in course structure. This suggests that we may be able to generate predictive models based upon early offerings of MOOCs and then use them on to enhance the later iterations. This will enable instructors to identify students who are likely to earn a certificate or to dropout in the first few weeks of the course and may be able to help or provide more support for the students in need.

One limitation of this work is that dropout is not pre-defined in the dataset, and we have no ground truth on the students' intentions when they quit. We therefore need to make assumptions when defining dropout, which can change the findings of the study. Our definitions of dropout, would count students with any kind of activity still engaged, thus if a student kept watching videos but not submitting assignments, they would not be counted as having dropped out. However, different assumptions on the definition of dropout might change the findings. Also, our analysis of the posts made by the central users is limited. Deeper study of the content in posts and replies, or whether they are on topic, can make the findings stronger.

One other limitation of our study was the imbalanced nature of our dataset. In both offerings, the majority of the students dropped out according to our definitions. In order to address this problem we randomly removed most of the dropped out students to balance this label as half true and half false. In future studies, more approaches should be tried to balance the dataset, and to include more variety of data while removing the duplicates.

7. ACKNOWLEDGMENTS

This work was supported by NSF grant #1418269: "Modeling Social Interaction & Performance in STEM Learning" Yoav Bergner, Ryan Baker, Danielle S. McNamara, & Tiffany Barnes Co-PIs.

8. REFERENCES

[1] J. M. L. Andres, R. S. Baker, G. Siemens, C. A. Spann, D. Gašević, and S. Crossley. Studying mooc completion at scale using the mooc replication framework. 2016.

[2] S. Boyer and K. Veeramachaneni. Transfer learning for predictive models in massive open online courses. In C. Conati, N. Heffernan, A. Mitrovic, and M. F. Verdejo, editors, *Artificial Intelligence in Education*, pages 54–63, Cham, 2015. Springer International Publishing.

[3] C. Brooks, C. Thompson, and S. Teasley. A time series interaction analysis method for building predictive models of learners using log data. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, LAK '15, pages 126–135, New York, NY, USA, 2015. ACM.

[4] R. Brown, C. Lynch, M. Eagle, J. Albert, T. Barnes, R. S. Baker, Y. Bergner, and D. S. McNamara. Good communities and bad communities: Does membership affect performance? In *Proceedings of the 8th International Conference on Educational Data Mining, EDM 2015, Madrid, Spain, June 26-29, 2015*, pages 612–613, 2015.

[5] R. Brown, C. Lynch, Y. Wang, M. Eagle, J. Albert, T. Barnes, R. S. Baker, Y. Bergner, and D. S. McNamara. Communities of performance & communities of preference. In *EDM (Workshops)*, 2015.

[6] Y. Chen and M. Zhang. Mooc student dropout: Pattern and prevention. In *Proceedings of the ACM Turing 50th Celebration Conference - China*, ACM TUR-C '17, pages 4:1–4:6, New York, NY, USA, 2017. ACM.

[7] M. Dash and H. Liu. Feature selection for classification. *Intelligent data analysis*, 1(3):131–156, 1997.

[8] J. E. Eckles and E. G. Stradley. A social network analysis of student retention using archival data. *Social Psychology of Education*, 15(2):165–180, 2012.

[9] M. Fei and D.-Y. Yeung. Temporal models for predicting student dropout in massive open online courses. In *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*, pages 256–263. IEEE, 2015.

[10] L. C. Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1978.

[11] N. Gitinabard, L. Xue, C. Lynch, S. Heckman, and T. Barnes. A social network analysis on blended courses. In *Proceedings of the 10th International Conference on Educational Data Mining, EDM 2017(Workshops), Wuhan, China, June 25-28, 2017*, 2017.

[12] C. Gütl, R. H. Rizzardini, V. Chang, and M. Morales. Attrition in mooc: Lessons learned from drop-out students. In *International Workshop on Learning Technology for Education in Cloud*, pages 37–48. Springer, 2014.

[13] S. Halawa, D. Greene, and J. Mitchell. Dropout prediction in moocs using learner activity features. *Experiences and best practices in and around MOOCs*, 7:3–12, 2014.

[14] S. Jiang, S. M. Fitzhugh, and M. Warschauer. Social positioning and performance in moocs. In *Workshop on Graph-Based Educational Data Mining*, volume 14, 2014.

[15] S. Jiang, A. Williams, K. Schenke, M. Warschauer, and D. O'dowd. Predicting mooc performance with week 1 behavior. In *Educational Data Mining 2014*, 2014.

[16] S. Joksimović, D. Gašević, V. Kovanović, B. E. Riecke, and M. Hatala. Social presence in online discussions as a process predictor of academic performance. *Journal of Computer Assisted Learning*, 31(6):638–654, 2015.

[17] K. Jordan. Initial trends in enrolment and completion of massive open online courses. *The International Review of Research in Open and Distributed Learning*, 15(1), 2014.

[18] C. Kartsonaki. Survival analysis. *Diagnostic Histopathology*, 22(7):263–270, 2016.

[19] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, Sept. 1999.

[20] 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.

[21] R. G. Miller Jr. *Survival analysis*, volume 66. John Wiley & Sons, 2011.

[22] B. Pursel, L. Zhang, K. Jablokow, G. Choi, and D. Velegol. Understanding mooc students: Motivations and behaviours indicative of mooc completion. *J. Comp. Assist. Learn.*, 32(3):202–217, June 2016.

[23] 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 *Proceedings of the first ACM conference on Learning@ scale conference*, pages 197–198. ACM, 2014.

[24] T. Sinha, N. Li, P. Jermann, and P. Dillenbourg. Capturing" attrition intensifying" structural traits from didactic interaction sequences of mooc learners. *arXiv preprint arXiv:1409.5887*, 2014.

[25] C. Taylor et al. *Stopout prediction in massive open online courses.* PhD thesis, Massachusetts Institute of Technology, 2014.

[26] A. Vihavainen, M. Luukkainen, and J. Kurhila. Using students' programming behavior to predict success in an introductory mathematics course. In *Educational Data Mining 2013*, 2013.

[27] Y. Wang. Mooc leaner motivation and learning pattern discovery. In *EDM*, pages 452–454, 2014.

[28] D. Yang, R. Kraut, and C. P. Rosé. Exploring the effect of student confusion in massive open online courses. *Journal of Educational Data Mining*, 8(1), 2016.

[29] D. Yang, T. Sinha, D. Adamson, and C. P. Rosé. Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In *Proceedings of the 2013 NIPS Data-driven education workshop*, volume 11, page 14, 2013.

[30] M. Zhu, Y. Bergner, Y. Zhang, R. Baker, Y. Wang, and L. Paquette. Longitudinal engagement, performance, and social connectivity: a mooc case study using exponential random graph models. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 223–230. ACM, 2016.

Predicting Student Performance Based on Online Study Habits: A Study of Blended Courses

Adithya Sheshadri, Niki Gitinabard, Collin F. Lynch, Tiffany Barnes, & Sarah Heckman
North Carolina State University, Raleigh, NC, USA
{ aseshad, ngitina, cflynch, tmbarnes, sarah_heckman }@ncsu.edu

ABSTRACT

Online tools provide unique access to research students' study habits and problem-solving behavior. In MOOCs, this online data can be used to inform instructors and to provide automatic guidance to students. However, these techniques may not apply in blended courses with face to face and online components. We report on a study of integrated user-system interaction logs from 3 computer science courses using four online systems: LMS, forum, version control, and homework system. Our results show that students rarely work across platforms in a single session, and that final class performance can be predicted from students' system use.

Keywords

logs, blended courses, MOOCs, study habits, predictive models

1. INTRODUCTION

Today, students enrolled in university courses, particularly those in STEM disciplines, supplement or even supplant class attendance with online materials such as Learning Management Systems (LMSs), discussion forums, intelligent tutoring systems, automatic graders, and homework helpers. TAs now offer online office hours even for local students and lecture videos are reviewed in lieu of note-taking. The goal of these tools is to foster a rich learning environment; to support good study habits; and to enable instructors to give and grade realistic activities such as collaborative learning and project development [10]. In addition to planning class lectures, course instructors now manage a constellation of online services which students can use, or not, at their own pace. In practical terms, many traditional face-to-face on campus courses are blended courses.

While these tools can be beneficial, blended courses normally use several different tools, requiring students to switch between different websites several times to access lecture notes, online discussions, and assignment submission systems. Even when the tools are linked through a single LMS such as Moodle, the need to transition from platform to platform can be challenging. In order to engage effectively with such hybrid materials, students need to develop good online study skills and need to effectively integrate information across platforms. While we have generally assumed that technically-savvy students have these skills, little work has been done to empirically evaluate how students use these platforms and whether or not their observed habits can be used to predict their performance.

While many of the systems used in blended courses are similar to those in Massive Open Online Courses (MOOC), studies of blended courses are limited. Prior work has mostly focused on single tools, and generally lack early prediction power [24, 18, 16, 1, 9, 19]. Prior research has suggested novel methods that can be used to predict student performance in MOOCs, based upon features extracted from students' interactions with different learning materials [5, 21, 14, 11]. Since not all of the students' learning activities can be monitored online, it is not certain whether the same methods can be applied [3].

In this paper, we present our work on the automatic analysis of students' study behaviors in blended courses. We focus on 3 Computer Science courses at North Carolina State University using 4 online platforms: Moodle, a large-scale LMS; Piazza, an online discussion forum for student questions; Github Enterprise, a project management platform for software development; and WebAssign, an online homework and automatic grading platform. Our goals in this work are: to *synthesize* data from these heterogeneous learning platforms; to *extract* meaningful student behaviors from the interaction data; and to *model* students' behaviors to predict their future performance and thus to provide a basis for automatic feedback or instructor support. We want to leverage this synthesized data to analyze not only *what* features of these online platforms students use but *when* they do so. This, in turn, can give us a deeper understanding of students' study habits and allow us to distinguish effective strategies from ineffective ones, facilitating automated support.

We address the following research questions: (Q1) Do students focus on one tool at a time or work across platforms?; (Q2) Do students' online study habits follow clear patterns based upon the course deadlines?; and (Q3) Can students' study habits be used to predict their final scores? We hypothesize that students tend to *silo* their work in one platform at a time, and that their tool use will predictably follow course deadlines. We also hypothesize that these patterns, or deviations from them, can be used to predict their overall performance across classes.

2. RELATED WORK

The use of online tools such as discussion forums, learning management systems (LMS), and online assignment submission systems in classrooms can provide researchers with more information on students' behavior than they can obtain through observation alone. MOOCs are attractive to researchers in part because they provide a *data choke-point* which highlights most relevant student interactions. While most of the data available in MOOCs is also available in blended courses, many of the findings on MOOCs have not been replicated in blended (face to face and online) courses in part due to a lack of available datasets and the fact that not all learning activities are tracked or logged. In this section, we first discuss some of the studies on MOOCs and performance prediction, then we discuss the research on blended courses.

There have been a number of studies of students' behavior in MOOCs and whether or not it is correlated with their overall performance. Seaton et al., analyzed students' use of course materials on an existing EdX MOOC with the goal of determining when and how the students attempt assignments and how they divide their time over the differing activities [20], finding that only 25% of the students attempt more than 5% of the assignments but those students account for 92% of the total time spent on the course. Sixty percent of the total time spent on the class was invested by the 6% of students who earned a certificate. There have been several attempts to predict student certification and dropout in MOOCs using features extracted from their online activities such as the number of videos watched in a week, the number of quiz or assignment attempts, the number of forum posts made per week, post length, relative time spent on assignments, and so on [17, 8, 4, 6]. Some researchers have gone further by defining study sessions for students, and using the sequence of students' access to the online material to make predictions on performance [2, 5, 14]. Amnueypornsakul et. al. defined active study sessions for each student and used the sequence of actions within sessions to define features such as length of the action sequence, the number of occurrences of each activity, and the number of Wiki page views [2]. Li et. al. applied the idea of N-grams to the sequence of student actions in a session and used those N-grams to predict the students' certification [14]. Brooks et. al. defined sessions with fixed duration such as 1 day, 3 days, 1 week, and 1 month throughout the semester and recorded students' activity within each time unit as a binary feature [5]. They defined N-grams on the sequences of features to make early and cross-class predictions of student dropout. Sinha et. al. added the concept of an interaction graph which connected students to the resources they accessed and found that a predictive model trained on the graph features can outperform N-gram based models for student behaviors [21].

These prediction methods, if applicable to blended courses, could help instructors to identify struggling students early in the semester for support. But it is still not clear how or if these behaviors can transfer from one domain to another. An et. al. tried replicating some of the predictive methods found in MOOCs to a blended course and found that those findings can be applied with some caveats as there were some changes needed in the design process for it to be applicable in other contexts [3]. For example, students' download activities were shown to cluster students into two categories of completing and auditing both in a MOOC and an online course, but this pattern was not visible in the blended course of their study. However, this clustering based upon assessment scores could identify some of the groups visible in MOOCs, also in the blended course.

Prior analyses of student behaviors in blended courses show that the overall level of activity increases when exam, quiz, or assignment deadlines are near at hand [16]. Analyses of students' login behaviors also show that the students' activities follow a predictable weekly cycle dropping on Saturdays and then rising on Sundays as they prepare for the week ahead [1]. Research has also shown that better performing students usually start and end their activities earlier than their lower performing peers [23] and that it is possible to use some student activity features to predict their performance [24, 18, 9, 1, 22]. Some examples of these features are attendance, emotions during lecture, number of assignments done, the time they took to do those assignments, number of posts on the discussion forum, prior scores, number of attempts, etc. Most of this prior work has been based upon *complete* datasets which represent all of the information obtained during a semester. Such models cannot therefore be used for early warnings or interventions. Munson et. al., by contrast, showed that features such as early scores, hours coding, error ratio, and file size can identify struggling students in the first three weeks of the class [15].

Class	Source	Total actions	Avg. per student	σ
DM 2013	Moodle	17148	166	88
	Piazza	2557	15	28
	WebAssign	265510	1062	201
DM 2015	Moodle	21972	80	59
	Piazza	2208	12	17
Java 2015	Moodle	101180	613	266
	Piazza	2556	20	25
	Github	31438	196	140

Table 1: Actions Taken by Platform

We focus on evaluating students' online tool use in terms of *sessions*, consecutive sequences of study actions that occur between breaks for food or sleep. Sessions have been previously used to analyze student behavior in MOOCs and in other cohesive online tools such as an LMS (e.g. [12]). Our work here extends that research by applying to heterogeneous data from blended courses where students can work across platforms and over longer periods of time with the goal of developing early predictors that can be used to identify high- or low-performing students in time for an intervention.

3. DATASETS

In order to address our research questions, we collected student data from three blended courses in Computer Science offered at North Carolina State University. Two are offerings of "Discrete Mathematics for Computer Science," (DM) from 2013 and 2015 respectively. The other was an offering of "Programming Concepts in Java" (Java) from 2015. Both courses are core courses for students majoring in CS and both are structured as blended courses. Students in both DM and Java use Moodle, an open-source LMS that is used for all courses at NC State University, and Piazza, an on-line Q&A platform for question answering that can be used for threaded discussions. The students in the DM classes use WebAssign and the students in the Java class used Github for assignment submissions.

We collected data from Piazza and Moodle for all these classes as well as data from the WebAssign system for the 2013 DM course. We also collected a complete record of students' Github commits from the 2015 Java course. The data was collected as web logs, database dumps, and in the case of WebAssign, via screen scraping. We then cleaned up the raw data, linked it across platforms, and anonymized it to produce a single coherent database for analysis.

For the purposes of this analysis, we focused solely on the students' actions and ignored actions by the course instructors. We also eliminated students who dropped out of the course (only possible during the first two weeks of the course) as well as students who did not get a grade (at most 20 students per class). Table 1 shows the total number of actions recorded for each tool along with the average number of actions per student removing zero values.

The final grades of all the students were provided to us by the instructors. In each case, the Grades are represented ordinally from **A**+to **F** from left to right. The relative distribution of student grades in the courses are shown in Figure 1. A majority of the students achieved good grades in each course offering (i.e. B or above) and fewer than 16% of them failed. We therefore based our predictive models on distinguishing students who achieved distinction in the course (A- to A+ grades) from those who did not. Due to the high proportion of distinction this yielded nearly balanced datasets.

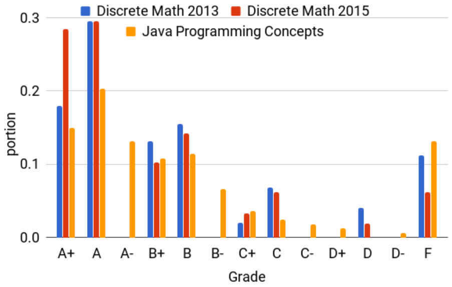


Figure 1: Grade Distribution for the Three Classes

3.1 Discrete Mathematics

Discrete Mathematics (DM) is a core CS course that introduces students to the basics of logic and proofs, set theory, and probability. Students typically enroll in the course during their second year and must pass it with a C or better to progress in the curriculum. Our data comes from the Fall 2013 and Fall 2015 offerings of the course. In both cases, the course was broken into two sections with two instructors and five shared Teaching Assistants (TAs). The 2013 offering had 250 students enrolled while the 2015 offering had 277. The average final grades in 2013 and 2015 were 89.5 and 86.9 out of 100 respectively. In both years the course had 10 homework assignments (30%), 5 lab assignments (10%) and 4 tests accounting for 60% of their final grades.

Students in the DM courses used WebAssign, an online homework platform. It is used to deliver, view, and grade student assignments. Assignments may be graded both manually and automatically. WebAssign is used in the DM course for weekly assignments linked via Moodle. We have access to WebAssign data for 2013 DM offering. Each submission shows a single attempt to answer a question in an assignment and provides information on the student making the submission, the time of the submission, the assignment, question information and the sub-question part being completed.

3.2 Programming Concepts in Java

Programming Concepts in Java is also a core CS course that covers software topics such as: system design and testing, encapsulation, polymorphism, finite-state automata, and linear data structures and recursion. Like the DM course it is offered to students during their second year and students must pass with a C or better. To obtain the letter grade earned, the students must obtain an average of 60% or better on the exams and assignments. In 2015, this class was structured into three sections with one instructor per section. One section was a pure distance education section with a completely separate student population. This was omitted from our analysis for the sake of consistency. Our dataset, therefore, covers 2 sections with 2 instructors, 9 teaching assistants, and 180 local students. The high TA to student ratio is due to the fact that this course involve a substantial coding project components and also the external funding supporting additional TAs. The course included 3 projects, 2 midterms, and one final exam and their final grades are generated based on the grades on all these activities. The average final grade for this class was 79.7.

Students in the Java course use GitHub, a version control system used widely in Software Engineering projects to allow users track changes to the code and share coding tasks within a team. Github is used in the Java class as a tool for individual and team projects, and also as an assignment submission system. The system is connected to an automatic test suite and students are graded based on their latest Github commits. Each record in our logs identifies the author, the number of changes to the code, and the time of submission.

4. SESSIONS

We aggregated the heterogeneous actions described above into a unified transaction log. This data consists of 285,465 transactions from the DM 2013 class, 24,180 transactions from the DM 2015 class, and 135,351 transactions from the Java class. As Table 1 shows, the lion's share of these transactions are WebAssign actions from Discrete Math 2013 and Moodle actions from the other two course offerings.

We divided the individual transactions into *sessions* representing contiguous sequences of student actions using data-driven cutoffs. Our goal in grouping the student actions was to better understand the students' online study habits and their longer-term strategies. Aggregating student actions into sessions is a nontrivial problem. Fixed time cutoffs can have incorrect edge cases and the time cutoff used can affect our final results. Kovanovic et. al., for example, evaluated the impact of different time cutoff strategies for a dataset extracted from a single LMS [12]. They found that there was no one best cutoff and recommended exploring the data to select a context-appropriate cutoff. Some techniques that have been used to estimate sessions are:

Fixed duration: Sessions can be defined based upon a fixed (often a priori) unit of time such as an hour or a day as in [5]. Sessions can also be defined by considering periods between assignment deadlines as the duration of the session.

Cutoff: Another method is using a fixed timeout or cutoff similar to Amnueypornsakul et. al. [2]. Here we group student actions into sections and separate sections when they go offline or pause for a set period of time. The consecutive actions between pauses then belong to a single session irrespective of its duration.

Navigation: In this approach, common in web-analysis, the actions themselves are analyzed to determine when a session has ended. If, for example, a user traverses to an unrelated content or engages in off-task browsing then we consider the session to be over.

Fixed duration sessions are unusable in this context as there is no clear time limit for the students' work to be completed. Navigation-based sessions are likewise unsuitable as our research questions are focused on all class activities and we do not have other external data that could be used to detect when students leave the systems. The selection of a cutoff is complicated by the fact that our data includes heterogeneous tasks. Some, such as answering a true/false question on WebAssign, are quick while others, such as composing Piazza posts, take time. In the absence of clear sign-off behavior, we chose to take a data-driven approach to selecting our cutoff values.

To that end, we began by plotting a histogram of the relative gaps between sequential actions on the different platforms. Most of the actions occurred quite close to one another with our histograms showing an exponential reduction in frequency from 1 to 201 seconds where it trails off. While informative, that analysis conflated two kinds of breaks, some where the student later

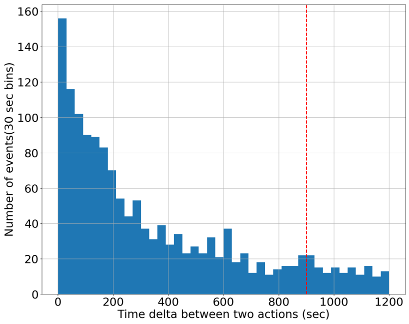


Figure 2: Histogram for Gap Length with Change of Platform for the 2013 Discrete Math Class.

returns to the same system (the most common case) with those where they shift to a different platform. We, therefore, plotted separate histograms for each type of gap, an example for gap length with cross platform activities is shown in Figure 2. The plot for actions on a single platform is similar but with shorter cut-off time visible. All three classes show a similar breakdown of gaps, thus we have only included the figure generated from the 2013 DM class here. As the figures illustrated, the gaps in which students change from one platform to another are generally longer and have longer tails. The within-platform gaps follow the same pattern as the full set with a sharp dropoff after 210 seconds. By contrast, the cross-platform gap has a mild dropoff after 600 seconds.

Given the variation of the data, it is not possible to define a single cutoff value that accurately captures all cases. We, therefore, opted to define two session types with two different cutoff values.

Browser Session: $m = 15$ minutes indicating a short break likely with the same browser open.

Study Session: $m = 40$ minutes indicating that student likely changed tasks or quit working entirely.

The *Browser Session* can be characterized as a case where the students are actively working on a single task with little change. This may include working on a multi-part WebAssign question, reading through materials on Moodle, or diagnosing an issue with their code with guidance from Piazza. Sessions of this type were comparatively short in duration. The *Study Session* by contrast allows for a much larger gap where students may shift from reading materials to answering questions or engaging in (online) discussions with their peers, and back again. This large cutoff was based upon the cross-platform breakdown and was in part intended to address our lack of data regarding the students' offline activities.

5. RESULTS & ANALYSIS

Table 2 presents basic statistics on the two types of sessions across the three classes. Because these sessions are defined by a time cutoff they have overlapping instances. Thus in DM 2013, 12,349 of the sessions were both study and browser sessions meaning that the gap between them and the neighboring sessions was over 40 minutes long and all of the internal gaps were less than 15 minutes. When analyzing the session duration, we found that almost half of the sessions (of both types) were comparatively short with the students making less than five actions. The relative frequency of the sessions drops quickly as the session length increases. Similar trends were exhibited when we examine the length of each session based on their duration. We also observe that the average duration of the sessions in DM 2015 dataset was drastically shorter than those in DM 2013 and Java 2015. This may be explained by the fact that the WebAssign records were present in DM 2013 while GitHub was included in Java 2015. This would give a more frequent and detailed picture of students' problem-solving. Removing Github activities from the Java class records resulted in a similar pattern of shorter sessions.

Class	Session	Count	Avg Duration	Homogeneous	Heterogeneous
DM 2013	Browser	17699	9 min	16892	777
	Study	14574	16 min	13668	906
DM 2015	Browser	10981	2 min	10963	18
	Study	10038	2 min	9994	44
Java 2015	Browser	28768	5 min	28645	123
	Study	25005	17 min	21932	223

Table 2: Information on Different Types of Sessions

5.1 Q1: Homogeneity

The browser and study sessions can be classified as heterogeneous and homogeneous sessions. *Heterogeneous sessions* occur when the student switches between platforms at least once during the session. *Homogeneous sessions* occur when no such change takes place. Table 2 presents a breakdown of the two types across the classes. As the table illustrates, in all of the classes more than 95.5% of the browser sessions are homogeneous as are more than 93.8% of the study sessions. These results are consistent with our hypothesis that students do not work across platforms but instead *silo* their activities working on one system at a time. This is true even with the long timeout for the study sessions. When they do transition from one platform to another it is largely a transition between Moodle, which links course schedules to assignments, and WebAssign, which allows them to complete their assignments in the DM 2013 class, or between Moodle and Github in the Java class.

5.2 Q2: Patterns

As noted above, the grade distribution for the courses is quite high. We therefore classified the students into one of three categories for analysis: *Distinction* (A+, A, or A-); *Pass* (B+, B-, B, C+, or C-); and *Fail* (D or F). We plotted the the frequency of individual browser sessions day by day over the course of the semester, and example of them for DM 2013 is shown in Figure 3. The red line indicates the Fail group, the blue line corresponds to Pass group and green represents the Distinction group. The vertical bars show the due dates for assignments and exams. As the plot illustrates, the number of sessions spike before each deadline for *all three* of the performance groups. A similar pattern was observed for the other classes, the frequency of the study sessions and also for the duration of both session types sessions. These results are consistent with our hypothesis and prior studies that students are deadline-driven even in blended courses. We also observe that the Fail group performs much fewer activities than the other two groups, getting close to zero in DM 2015 class. This shows that most of the actions the Fail group performed were WebAssign actions, which we do not analyze for the 2015 class.

5.3 Q3: Prediction

As Figure 3 and the other class plots illustrated, all three performance groups in all three classes followed a similar pattern. All of the groups have irregular activity patterns and all of them see spikes prior to each of the deadlines and exams. Yet there are important differences among the groups. As a group, the Distinction students were always active, with the number of active sessions rarely if ever, reaching zero. The Fail students, by contrast, were frequently inactive as a group with long periods where no fail

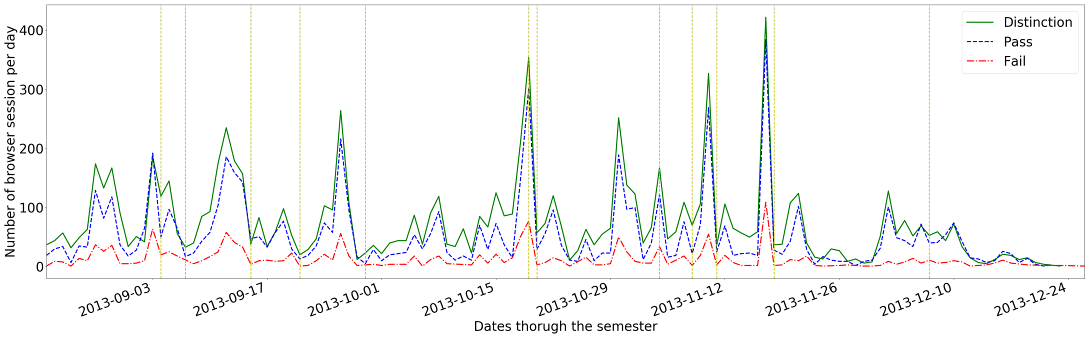


Figure 3: Frequency of Browser Sessions with Assignment Deadlines and Test Dates for Discrete Math 2013

Table 3: Kruskal-Wallis P-values for Succeed/Fail Classification, Values Less than 0.05 Are Illustrates in Bold

Parameter	Pre Test 1			Pre Test 2			Full Semester		
	DM	DM	Java	DM	DM	Java	DM	DM	Java
	2013	2015	2015	2013	2015	2015	2013	2015	2015
Avg Gap	0.15	**0.02**	**0.00**	**0.00**	**0.03**	0.12	0.10	**0.04**	**0.02**
Num Sessions	**0.01**	**0.03**	**0.00**	**0.00**	**0.02**	0.06	**0.00**	**0.01**	**0.04**
Pratio	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.04**	**0.00**	**0.00**	0.08
Total Time	**0.00**	0.13	**0.04**	**0.00**	0.23	0.25	**0.01**	0.20	0.08
Consistency	0.07	**0.01**	**0.00**	**0.00**	**0.03**	0.10	0.08	**0.03**	**0.03**
Total Actions	**0.00**	0.14	**0.02**	**0.00**	0.08	0.14	**0.01**	0.17	0.19
Piazza Questions	**0.00**	**0.00**	**0.04**	**0.00**	**0.00**	0.08	**0.00**	**0.00**	0.10
Piazza Answers	**0.00**	**0.00**	**0.04**	**0.00**	**0.00**	0.09	**0.00**	**0.00**	0.11
MultipleAttempts	**0.00**			**0.00**			**0.00**		

student was active at all. The Pass students, by contrast, occupy an interesting middle ground with less consistent activity than the Distinction group but far more than the Fail group. This suggests that students who succeed in the courses are broadly more consistent and engage in online activity at regular intervals. However, all the students are given to cramming (spending much more time on classes right before tests) with the better students cramming as much or more than their peers. These results also indicate that the relative gap between sessions may be a significant predictor of students' individual performance.

Based upon those results we then identified a set of 16 session features for deeper analysis: Sessions: count, avg actions, total actions; avg duration, time=count* avg duration, avg gap, consistency, homogeneous, heterogeneous; Piazza: questions, answers, ratio of sessions containing Piazza activity; Webassign (DM2013 only): parts submitted, first attempts, multiple attempts; Performance (distinction, pass, fail);

$$Consistency = AvgGap * (max(SessionsCount) - SessionsCount)$$

We reclassified the students into two categories, *Distinction* and *Non-distinction* (Fail and Pass students) and applied the Kruskal-Wallis (KW) test, a non-parametric analogue to the Analysis of Variance test (ANOVA) [13], to determine whether or not any of these features are significantly correlated with having high performance in the course. The Kruskal-Wallis test is a good choice in this context because it does not assume normally-distributed data. Table 3 lists the features that were significant among the groups. As that table illustrates 9 of the features were statistically significant predictors of whether or not the students would pass the course. Crucially, some of these features were significant predictors of student performance even when we restricted our focus to data from the first half of the semester or to the first quarter (3 weeks). It is not surprising that the significant features differed between the classes given the absence of WebAssign data from two of the courses and the use of GitHub in the Java class. It is interesting however, that even without including WebAssign data source in the Discrete Math 2015 class, we can observe significant correlations between the defined features and performance. Our results illustrate that in all the classes, most of these features are significant early in the semester and the sign of the coefficients do not change in different time frames and across classes.

We extended this analysis by testing whether or not these values correlated with students' final grades using Kendall's τ a non-parametric correlation coefficient that works well with small sample sizes and is robust in the presence of ties [7]. Table 4 lists the τ coefficient and p-values for the features that were significantly correlated with the students' final grades. As the table illustrates, most of the features were significantly correlated with the final grades in all the classes, though the coefficients were small. Moreover, the direction of the correlations did not change over the course of the semester. In the 2015 Java dataset however, none of the features were correlated with the data before test 2. It is not entirely clear why the results are so different for this course. The gap may be explained by a change in the course activities in the second part of the class that is not adequately reflected in our dataset.

These results are largely consistent with our hypotheses, particularly for the DM offerings. We can use individual variables to predict whether or not the students will pass the course with high performance, based upon some of their per-session features. More importantly, we can do so based on the first few weeks of the course. Thus, it is possible to identify students who may be in need of support early when there is still time to change student behaviors.

5.4 Predictive Models

We expanded on these results by training predictive classifiers for the students' course performance based upon the correlated features. For the models including data after test 1, we included the test 1 grade and for the model based on all semester data, we included both tests 1 and 2 grades. We used logistic regression, decision tree, and K Nearest Neighbor classifiers to predict student performance using data generated before test 1, before test 2, and over the course of the entire semester. We generated a model for each course and subset, which could classify students into Distinction/Non-Distinction groups. The $F1$ scores for these models are shown in Table 5. While the best performing classifier varies among different classes and subsets, the best models based

	Data before Test 1			Data before Test 2			Total Semester Data		
	DM 2013	DM 2015	Java 2015	DM 2013	DM 2015	Java 2015	DM 2013	DM 2015	Java 2015
Avg Gap	-0.075	-0.1209**	-0.1334*	-0.1411**	-0.1171**	-0.0994	-0.0917*	-0.1212**	-0.1629**
Num Sessions	0.1413**	0.1445***	0.1548**	0.1833***	0.1386***	0.0913	0.2121***	0.1484***	0.1427**
Pratio	0.2461***	0.2036***	0.1192*	0.2502***	0.2196***	0.0712	0.3090***	0.2299***	0.0716
Total Time	0.1638***	0.1382***	0.1205*	0.1858***	0.1205**	0.0795	0.1759***	0.1210**	0.1647**
Consistency	-0.0879*	-0.1253**	-0.1374**	-0.1456***	-0.1197**	-0.099	-0.0996*	-0.1233**	-0.1535**
Total Actions	0.1782***	0.1225**	0.1154*	0.2038***	0.1265**	0.076	0.1648***	0.1131**	0.1319*
Test 1							0.5141***	0.5216***	0.5141***
Test 2							0.4783***	0.6582***	0.4783***

Table 4: Kendall's τ for the Defined Parameters and the Final Course Outcome (*: $p<0.05$, **: $p<0.01$, ***: $p<0.001$)

	Before Test 1			Before Test 2			Full Semester		
Parameter	DM 2013	DM 2015	Java 2015	DM 2013	DM 2015	Java 2015	DM 2013	DM 2015	Java 2015
Decision Tree	0.5432	0.3881	0.5172	0.6750	0.6032	0.7500	0.6774	0.6769	0.7273
KNN	0.5067	0.4815	0.3922	0.5352	0.4444	0.5333	0.7164	0.7857	0.7778
Logistic Regression	0.6364	0.3182	0.6885	0.6333	0.5763	0.7600	0.6452	0.8077	0.6939

Table 5: F-Measure Performance for Distinction Group Prediction

upon full semester or even before test 2 activities for all the classes performed with decent accuracy.

6. CONCLUSIONS & FUTURE WORK

Blended courses which pair face to face lectures with multiple distinct online learning platforms are increasingly the norm, particularly in STEM domains. Our goal in this paper was to determine whether or not it is possible to automatically analyze students' online study behaviors to identify good and poor study habits with the goal of supporting instructors and of providing automated guidance. In particular, we sought to address the following three research questions: (Q1) When working with online resources do students focus on a one tool at a time or do they work across platforms?; (Q2) Do students' online study habits follow clear patterns based upon the course deadlines?; and (Q3) Can students' observed study habits be used to predict their final scores?

In order to address these questions, we collected data from three CS course offerings at North Carolina State University. Two were separate instances of the same course while the other represented a different topic and instructor. All three courses used a range of online tools, we collected data from four critical ones: a shared LMS, an online discussion forum, an online homework platform, and a version control system. We then developed methods to synthesize this heterogeneous student data across the platforms and examined students' individual study actions grouping them into study and browser sessions using empirical cutoffs. We then grouped students into separate categories based on their performance and analyzed the pattern of sessions observed for each group. And finally, we identified key features of the students' online habits, assessed the relative correlation of those habits with their final performance, and trained classifiers to predict their performance.

In each case, we found that the data was consistent with our hypotheses. Students in the course typically siloed their work on the platforms (homogeneous sessions) and rarely, if ever, used two or more platforms in a single session. The students' study and browser sessions spiked in advance of each course deadline or test and dropped precipitously afterward. This pattern was consistent for these undergraduate students regardless of their final performance.. And finally, we found that the students' study habits did differ based upon their level of performance and that key features of the study habits were significantly correlated with the students' performance and final grades. Moreover, some of these correlations were observed even in the first few weeks of the course, at a time when change is still possible. The features identified can be used to construct successful classifiers to predict performance and the individual features (e.g. average gap between sessions), lend themselves to clear direct feedback.

Prior researchers have shown that it is possible to analyze students' actions on MOOCs to predict their ultimate performance in the course. In MOOCs, we have a data choke point that yields largely complete records of students' course activities. In blended courses, however, we lack a complete data picture as the students still engage in face-to-face lectures, visit office hours, and meet directly to discuss materials, or to exchange solutions. In spite of this incomplete information we have shown that it is possible to analyze students' behaviors to derive pedagogically relevant information that can be used to support instructors or to provide automated guidance. While the induced classifiers are not perfect, and while they depend upon some crucial design decisions such as the session selection, they still have the potential to provide real benefits in everyday classrooms.

In the near term, we plan to extend this work by incorporating additional data that was unavailable for our present analysis such as records from Jenkins, an automatic test suite. We also plan to investigate other mechanisms to detect off-task behavior and to estimate time on task that are sensitive to the specific actions being taken. In longer-term work, we plan to develop a centralized platform for automatically logging and integrating data from heterogeneous sources to provide automatic strategic feedback. It is our hypothesis that regular feedback from a virtual "study buddy" can be useful in encouraging students to develop better work habits even with a relatively low rate of guidance.

7. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1418269: "Modeling Social Interaction & Performance in STEM Learning" Yoav Bergner, Ryan Baker, Danielle S. McNamara, & Tiffany Barnes Co-PIs, and by a Google CS Capacity award, and an NCSU DELTA Course Redesign Grant.

8. REFERENCES

[1] L. Agnihotri, A. Aghababyan, S. Mojarad, M. Riedesel, and A. Essa. Mining login data for actionable student insight. In *Proc. 8th International Conference on Educational Data Mining*, 2015.

[2] 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, 2014.

[3] T.-S. An, C. Krauss, and A. Merceron. Can typical behaviors identified in moocs be discovered in other courses? In *Proceedings of The 10th International Conference on Educational Data Mining (EDM 2017)*, pages 25–28, 2017.

[4] J. M. L. Andres, R. S. Baker, G. Siemens, C. A. Spann, D. Gašević, and S. Crossley. Studying mooc completion at scale using the mooc replication framework. 2016.

[5] C. Brooks, C. Thompson, and S. Teasley. A time series interaction analysis method for building predictive models of learners using log data. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, LAK '15, pages 126–135, New York, NY, USA, 2015. ACM.

[6] Y. Chen and M. Zhang. Mooc student dropout: Pattern and prevention. In *Proceedings of the ACM Turing 50th Celebration Conference - China*, ACM TUR-C '17, pages 4:1–4:6, New York, NY, USA, 2017. ACM.

[7] P. Dalgaard. *Introductory statistics with R*. Springer Science & Business Media, 2008.

[8] M. Fei and D.-Y. Yeung. Temporal models for predicting student dropout in massive open online courses. In *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*, pages 256–263. IEEE, 2015.

[9] N. Gitinabard, L. Xue, C. Lynch, S. Heckman, and T. Barnes. A social network analysis on blended courses. In *Proceedings of the 10th International Conference on Educational Data Mining, EDM 2017(Workshops), Wuhan, China, June 25-28, 2017*, 2017.

[10] C. R. Graham. Blended learning systems. *The handbook of blended learning*, pages 3–21, 2006.

[11] 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.

[12] V. Kovanovic, D. Gasevic, S. Dawson, S. Joksimovic, R. S. Baker, and M. Hatala. Penetrating the black box of time-on-task estimation. In J. Baron, G. Lynch, N. Maziarz, P. Blikstein, A. Merceron, and G. Siemens, editors, *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge, LAK '15, Poughkeepsie, NY, USA, March 16-20, 2015*, pages 184–193. ACM, 2015.

[13] 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.

[14] X. Li, T. Wang, and H. Wang. Exploring n-gram features in clickstream data for mooc learning achievement prediction. In *International Conference on Database Systems for Advanced Applications*, pages 328–339. Springer, 2017.

[15] J. P. Munson and J. P. Zitovsky. Models for early identification of struggling novice programmers. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, pages 699–704. ACM, 2018.

[16] J. Park, K. Denaro, F. Rodriguez, P. Smyth, and M. Warschauer. Detecting changes in student behavior from clickstream data. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, LAK '17, pages 21–30, New York, NY, USA, 2017. ACM.

[17] B. Pursel, L. Zhang, K. Jablokow, G. Choi, and D. Velegol. Understanding mooc students: Motivations and behaviours indicative of mooc completion. *J. Comp. Assist. Learn.*, 32(3):202–217, June 2016.

[18] C. Romero, P. G. Espejo, A. Zafra, J. R. Romero, and S. Ventura. Web usage mining for predicting final marks of students that use moodle courses. *Computer Applications in Engineering Education*, 21(1):135–146, 2013.

[19] S. Ruiz, M. Urretavizcaya, and I. Fernández-Castro. Predicting students' outcome by interaction monitoring.

[20] 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.

[21] T. Sinha, N. Li, P. Jermann, and P. Dillenbourg. Capturing" attrition intensifying" structural traits from didactic interaction sequences of mooc learners. *arXiv preprint arXiv:1409.5887*, 2014.

[22] J. Spacco, P. Denny, B. Richards, D. Babcock, D. Hovemeyer, J. Moscola, and R. Duvall. Analyzing student work patterns using programming exercise data. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, SIGCSE '15, pages 18–23, New York, NY, USA, 2015. ACM.

[23] S. Willman, R. Lindén, E. Kaila, T. Rajala, M.-J. Laakso, and T. Salakoski. On study habits on an introductory course on programming. *Computer Science Education*, 25(3):276–291, 2015.

[24] A. Zafra and S. Ventura. Multi-instance genetic programming for predicting student performance in web based educational environments. *Applied Soft Computing*, 12(8):2693–2706, 2012.

Analyzing the relative learning benefits of completing required activities and optional readings in online courses

Paulo F. Carvalho
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
pcarvalh@cs.cmu.edu

Min Gao
Beijing Normal University
19 Xin-jie-kou Wai St,
Beijing 100875, P. R. China
bnugm2014@163.com

Benjamin A. Motz
Indiana University
1101 E. 10th St.
Bloomington, IN 47405
bmotz@indiana.edu

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

ABSTRACT

Students who actively engage with learning materials, for example by completing more practice activities, show better learning outcomes. A straightforward step to stimulate this desirable behavior is to require students to complete activities and downplay the role of reading materials. However, this approach might have undesirable consequences, such as inflating the number of activities completed in a short period of time until maximum performance is achieved ("gaming the system"). In this paper, we analyze the relative benefits of completing activities vs. readings for learning outcomes in an online course that required students to perform practice activities. The results show that students who read more pages have better learning outcomes than students who completed more activities. This pattern of results holds even when considering different measures of active engagement but is reversed when considering only activities classified as effective active engagement by a "gaming behavior" classifier. Overall, these results suggest that, when completing activities is required, students benefit from complementing the activities with optional readings. One possibility is that completing optional readings can be an active learning activity in itself, driven by students who are going beyond the minimum requirement, and actively seeking further information and robust feedback that complements the activities.

Keywords

Active learning; online learning; "game the system" classifiers

1. INTRODUCTION

Students learn better when they engage in active learning [11,19]. Yet, much instructional practice emphasizes passive learning such as reading text, attending lectures, and watching videos. Contrary to evidence of the clear benefits of active learning, students (and a surprisingly high number of instructors) feel that passive strategies such as re-reading are useful study methods [16]. This disconnect between evidence and practice highlights the need to develop active learning practices that are grounded in empirical evidence and can support effective learning. In this paper we investigate the positive benefits of active learning in an online course and the effect of encouraging students to engage in pre-determined active learning activities.

Online courses might, by their nature, lead to fewer active learning practices. For example, online courses often rely on text and videos to convey information, typical passive learning practices. However, although video- or text-based online courses are common, previous work by Koedinger and collaborators has suggested that greater engagement with practice activities in online courses is a better predictor of improved learning than greater engagement with video or text materials [7,13,14]. In light of this research, one suggestion would be that more activities should be included in online courses, and students should be encouraged to complete them. However, two problems arise from trying to implement this suggestion: how to encourage students to complete activities and what type of activities to use.

Effective self-regulation skills play an important role for successful learning in in-person instruction [5], as well as in online courses [4,12]. With the added autonomy afforded by online courses compared to in-person instruction, students who lack appropriate self-regulation skills or try to complete the course with the minimum amount of time and effort might not perform as well. Thus, it is important to encourage students who might not otherwise engage in active learning to do so [5], both because it might be more time consuming and effortful than passive learning techniques but also because engaging in active learning stimulates self-regulation and accurate learning calibration [10]. One straightforward way to do so in online courses is to include multiple practice activities in each online lesson and make performance in the activities count towards the students' grade. This suggestion is not without its challenges, however. While this approach might encourage students who otherwise would not complete the activities to do so, it might be problematic if regulating one's own activities is a critical ingredient in the learning process. Indeed, previous research on other cognitive approaches to improve learning, have repeatedly shown a difference in outcomes between when students are in control of their study and when they are not [6,8]. Another issue is related to "gaming the system" behaviors. Making activity completion explicitly related to grade outcomes, might lead students to attempt to exploit the activities not as learning devices, but a way to quickly achieve better grades [1,2].

There is also the issue of how the activities should be designed. Previous research investigating the positive effect of completing more activities in online courses looked at courses using activities that not only were optional, but also included extensive feedback, both for correct and incorrect responses. It is possible that the characteristics of the activity used play a role on whether they contribute to improved learning [15].

With these questions in mind, in the current study we investigated the relative benefits of completing activities and reading textbook materials for learning outcomes in an online course. The main research questions were (1) whether completing more practice activities would contribute to better learning outcomes than accessing more textbook pages when students are required to complete the activities and are provided minimal feedback in the activities, and (2) whether we could detect students' active engagement in the activities and distinguish it from 'gaming the system' behaviors such as completing the same activity multiple times quickly until a high score was achieved.

We use data from an exclusively online course taught at Indiana University. This course had a few characteristics that made it particularly relevant for the current research questions: (1) it included many practice activities in each unit, (2) the activities were required, graded and made up a large part of the students' final grade, but (3) students were allowed to complete the activities as many times as desired, (4) the activities included only correctness feedback, and (5) the textbook materials were separated from the rest of the course materials.

We start by analyzing the relative benefits of completing more activities vs. accessing more textbook pages as in previous research. Next, we investigate possible markers of "active learning" engagement that might influence the relative benefit of completing more activities on learning outcomes that help identify behaviors to use in the classifiers. Finally, we developed two classifiers to detect, among all activity completion attempts, which ones might involve "active learning" behaviors, and which might involve "gaming the system" behaviors. We then use measures derived from these classifiers to evaluate the relative benefits of more active completions of activities vs. accessing more textbook pages.

2. DATA AND METHODS

We used data from two semesters of an online introductory psychology course at Indiana University (N = 247 and N = 492, respectively). All students enrolled in the course were undergraduate students at one of the campuses of Indiana University taking the course for credit. All students' rights as research participants were protected under a protocol approved by the local review board and were informed in the course syllabus that their data would be analyzed.

2.1 Course Description

Table 1. Number of assigned activities and textbook readings available in Units 2-7.

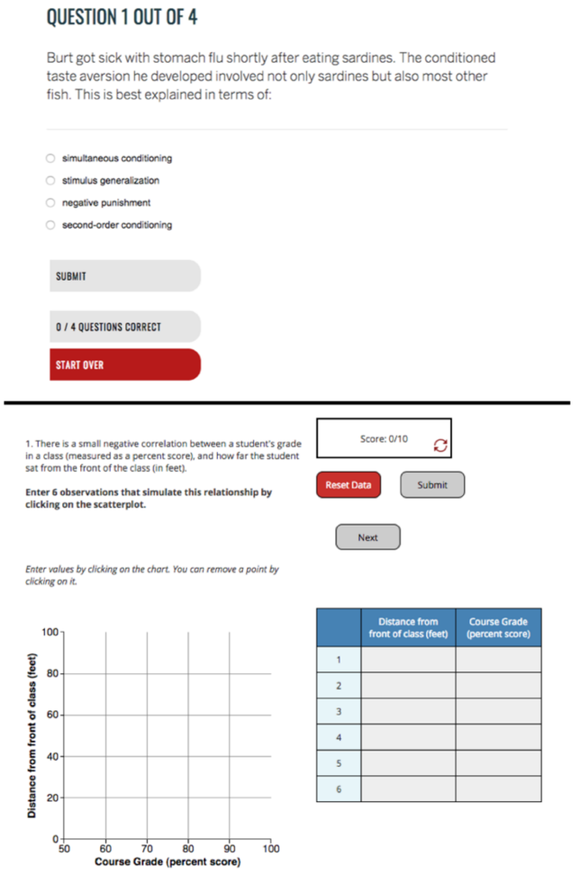
Unit	Number of lesson activities	Number of Textbook readings
2 – Methods	34	68
3 – Neuroscience	24	46
4 – Perception	30	43
5 – Memory	19	37
6 – Learning	20	44
7 - Cognition	18	36

The course was developed by the third author and delivered through Canvas. The course had seven units, but the first unit was purely introductory (there was no quiz at the end of the first unit) and is not included in the present analysis, leaving six content units for the current study (listed in Table 1). All units started with a short video from the instructor presenting an overview of the main topics of the unit. Moreover, every unit contained a different number of lessons, and within each lesson a different number of pages, each dedicated to a sub-topic. Every page contained an abbreviated summary of the main points of the sub-topic, links to the relevant readings of the online textbook, and lesson activities. Some pages also included videos and demonstrations. The number of lesson activities and textbook pages varied from unit to unit (see Table 1).

2.1.1 Lesson activities

Students were required to complete all the practice activities within the lessons of all units, using a custom LTI-based assessment platform installed in Canvas (Quick Check; https://github.com/IUeDS/quickcheck). Performance on these activities accounted for 45% of the students' final grade. Lessons were scheduled, and activities had to be completed within the scheduled time for each lesson. Students were allowed to complete the activities as many times as desired before or after the lesson completion deadline. Only their highest score before the lesson completion deadline was considered for their grade, and activities completed after the deadline never counted toward the students' grade. The lowest four aggregate lesson scores were automatically dropped. Aggregate lesson scores indicate the scores of all activities in the same lesson.

Lesson activities covered the content of the specific lesson they were assigned to and varied in format across lessons, including, for example, multiple-choice and graph interpretation activities. An example of two activities is included in Figure 1.



QUESTION 1 OUT OF 4

Burt got sick with stomach flu shortly after eating sardines. The conditioned taste aversion he developed involved not only sardines but also most other fish. This is best explained in terms of:

○ simultaneous conditioning
○ stimulus generalization
○ negative punishment
○ second-order conditioning

0 / 4 QUESTIONS CORRECT

1. There is a small negative correlation between a student's grade in a class (measured as a percent score), and how far the student sat from the front of the class (in feet).

Enter 6 observations that simulate this relationship by clicking on the scatterplot.

Score: 0/10

Enter values by clicking on the chart. You can remove a point by clicking on it.

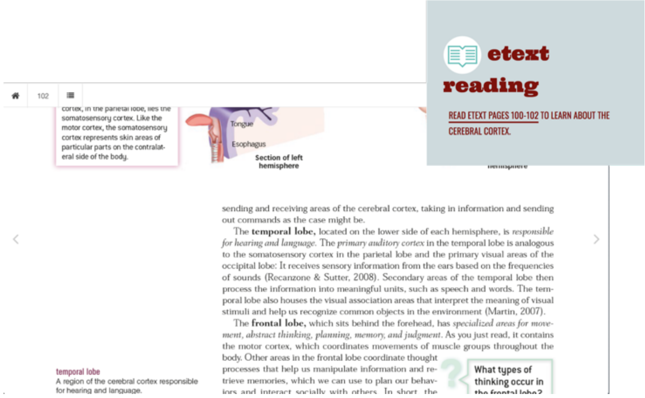
	Distance from front of class (feet)	Course Grade (percent score)
1		
2		
3		
4		
5		
6		

SUBMIT
START OVER
Reset Data
Submit
Next
100
80
60
40
20
0
Distance from front of class (feet)
50
60
70
80
90
100
Course Grade (percent score)

Figure 1. Examples of two lesson activities.

2.1.2 *Textbook readings*

The course used an online version of a commercially available Introductory Psychology textbook through the Unizin platform (eText). All students had access to the eText as part of their enrollment in the course. The relevant pages of the textbook for each topic covered in each page of every lesson across all units was provided in the course (see Figure 2 for an image). Students could access the eText at any point, including during the exams. Importantly, reading the eText was not incentivized or rewarded with points.



etext
reading
102
Tongue
Esophagus
Section of left hemisphere
hemisphere
What types of thinking occur in the frontal lobe?

READ ETEXT PAGES 100-102 TO LEARN ABOUT THE CEREBRAL CORTEX.

cortex, in the parietal lobe, lies the somatosensory cortex. Like the motor cortex, the somatosensory cortex represents skin areas of particular parts on the contralateral side of the body.

sending and receiving areas of the cerebral cortex, taking in information and sending out commands as the case might be.

The **temporal lobe,** located on the lower side of each hemisphere, is *responsible for hearing and language*. The *primary auditory cortex* in the temporal lobe is analogous to the somatosensory cortex in the parietal lobe and the primary visual areas of the occipital lobe: It receives sensory information from the ears based on the frequencies of sounds (Recanzone & Sutter, 2008). Secondary areas of the temporal lobe then process the information into meaningful units, such as speech and words. The temporal lobe also houses the visual association areas that interpret the meaning of visual stimuli and help us recognize common objects in the environment (Martin, 2007).

The **frontal lobe,** which sits behind the forehead, has *specialized areas for movement, abstract thinking, planning, memory, and judgment*. As you just read, it contains the motor cortex, which coordinates movements of muscle groups throughout the body. Other areas in the frontal lobe coordinate thought processes that help us manipulate information and retrieve memories, which we can use to plan our behaviors and interact socially with others. In short, the

temporal lobe
A region of the cerebral cortex responsible for hearing and language.

Figure 2. Example of link to eText pages on lesson (top right) and corresponding eText page in new window (bottom).

2.1.3 *Quizzes*

At the end of every unit, students completed a timed quiz online. Students could only attempt the quiz once within the time-frame allotted. Quizzes included a series of multiple-choice questions randomly chosen for each student from a larger pool. Quizzes accounted for 40% of the student's final grade and the lowest quiz grade was automatically dropped.

2.1.4 *Reflection activities*

Finally, students also completed a reflection activity for each unit. These activities were a writing assignment designed to help students think about the course materials for that unit in more depth. These assignments were due when the quiz for the unit became available and accounted for 15% of the students' final grade. The lowest score was automatically dropped.

2.2 Data

Detailed logged information was collected for this course. We analyzed information regarding when each lesson activity was attempted and how many times, how many eText pages were accessed and when, as well as scores on the lesson activities and the quizzes. The logged information allowed us to determine how long students took completing activities, but not how long they spent reading.

2.3 Model building

In order to compare the relative effect of different student behavior we normalized all measures by converting them to z-scores. Unless otherwise stated, we used mixed effects regression models to investigate the effect of different student behaviors on quiz scores. The baseline model included number of activities completed and the number of eText pages accessed. We predict quiz performance for each quiz, considering only behaviors that took place *before* the quiz was made available to the students:

$$\text{ZquizScores} \sim \text{Zactivities} + \text{Zpages} + (1|\text{student}) + (1|\text{quiz}) \quad (1)$$

This base model includes activities completed before the corresponding due date or after the due date as long as it was before the start of the quiz period. Considering only activities completed before the corresponding due date does not change any of the result patterns reported here. To help establish potential causal relations, we also ran the same baseline model predicting quiz grades using only behaviors that took place *after* the quiz was made available. We included student and quiz number as random effects in all models. We extracted different student behavior features and added them to the baseline model to infer the relative benefit of doing and reading using different properties of doing (e.g., time and accuracy). We use *chi-square* to compare models.

In addition, we developed two different classifiers to identify active engagement with the activities and discriminate it from possible "gaming the system" behaviors by the students (see below for details). We then include the 'gamer' classifier as an added predictor to the baseline model.

3. RESULTS

We started by running all analyses separately for each semester. All patterns were similar across both datasets; thus, we combined the two datasets into a single dataset for all analyses reported below. For brevity, we focus only on quiz performance as outcome measure, a similar pattern of results was found when considering the reflection activities as outcome measure.

3.1 Description of main variables

3.1.1 *Lesson activities*

Students completed an average of 74 activities before the quiz (Median = 71, SD = 40), and took on average 201 minutes (Median = 114, SD = 246) doing so. Only an average of 22% of these activities were completed after the activity due date but before the quiz, therefore in all subsequent analyses we consider any activity completed before the start of the quiz, regardless of the activity specific due date. After the corresponding quiz start date, students completed an average of 17 activities (Median = 5, SD = 25), taking on average 20 minutes doing so (Median = 0, SD = 61).

3.1.2 *eText*

Students opened an average of 22.5 eText pages before the corresponding quiz (Median = 5, SD = 35) and 12 pages after the corresponding quiz was made available (Median = 5, SD = 17).

3.1.3 *Quizzes*

The mean quiz score was 23.88 (Median = 25, SD = 4.80) out of 30 possible points. The distribution of quiz scores as a function of number of activities completed and number of pages opened before the quiz is presented in Figure 4.

3.2 Base models: Relative benefit of doing and reading before the quiz start date

3.2.1 *Behavior before the quiz start date*

Accessing more eText pages before the quiz being made available predicts better quiz performance, $\beta = 0.14$, t (3689) = 8.07, $p <$ .0001. Conversely, completing more activities before the quiz was made available predicts worse quiz performance, $\beta = -0.04$, t (3733) = -2.06, $p = .039$.

Overall, we do not see a "doer effect", i.e., that completing more practice activities improves learning outcomes to a larger degree than completing more readings.

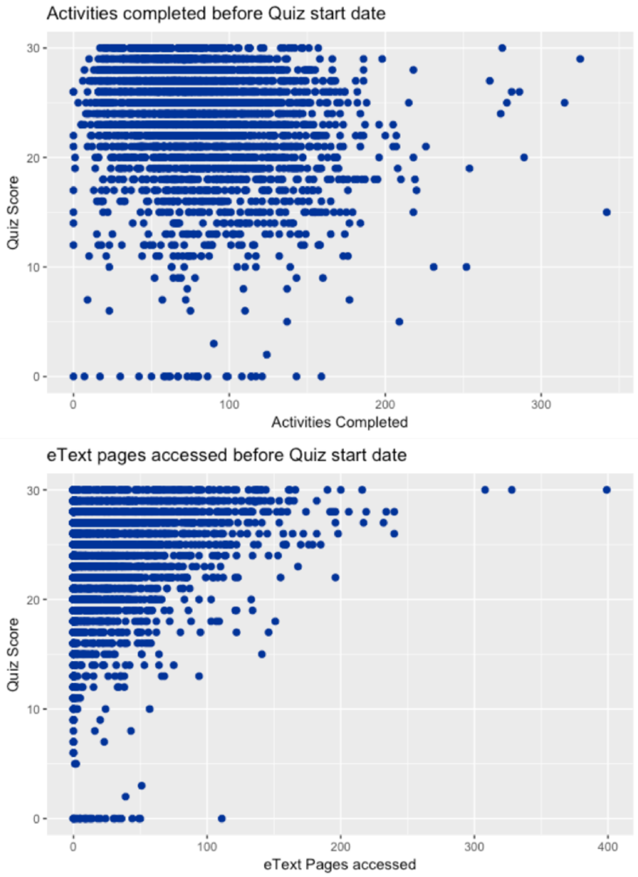


Figure 3. Distribution of quiz scores as a function of number of lesson activities (top panel) and eText pages (bottom panel) accessed before the quiz start date.

3.2.2 Behavior after quiz start date

Accessing more eText pages after the quiz was made available predicts better quiz performance, $\beta = 0.05$, t (3590) = 2.94, $p < .0001$, potentially because students were using the eText to complete the quiz. Completing more activities after the quiz was made available also predicts better quiz performance, $\beta = 0.18$, t (3810) = 12.38, $p < .0001$.

Thus, completing more activities after the quiz was made available had a larger effect on outcomes than accessing more eText pages, contrary to what we saw when analyzing behaviors before the quiz was made available.

3.3 Time and performance models

The learning benefit of completing more activities is likely to be connected with active engagement with the activities. However, the requirement to complete activities and the fact that performance on these activities directly affected students' grades might have led students to complete the activities multiple times in quick succession for maximum performance (a "gaming the system" behavior). This is potentially a different type of activity engagement that would not lead to a doer effect. To test this hypothesis, we created models that include measures potentially more related to active engagement: (a) time working on activities, (b) average performance across all activity attempts, and (c) best performance weighted by number of activity attempts. We compare models including each of these measures as added predictors with the baseline model for behaviors before the quiz described above.

3.3.1 Time working on activities

Spending more time working on the activities before the quiz has a positive impact on quiz performance, $\beta = 0.09$, t (3818) = 5.47, $p < .0001$. Moreover, compared to the baseline model, the activity time model provided a significantly better fit to the data, $\chi^2 = 29.34$, $p < .0001$ (see Table 2).

3.3.2 Average performance on activities

Higher average performance on the activities completed before the quiz is also related to higher quiz performance, $\beta = 0.04$, t (3796) = 2.604, p = .009. Compared to the baseline model, the activity performance model provided a significantly better fit to the data, $\chi^2 = 6.64$, $p = .01$ (see Table 2).

3.3.3 Number-weighted best performance

Only the highest score across all attempts was considered for student final grade. Therefore, it is likely that students who achieved higher scores with less attempts were more actively engaged in the activities than students who achieved higher scores with more attempts. The latter group was likely to be attempting to achieve a high score by completing the activity multiple times without attending to the actual question or feedback. Achieving highest scores in less attempts predicted better quiz results, $\beta = 0.04$, t (3365) = 3.10, p = .002. This model also provides a significantly better fit to the data compared to the baseline model, $\chi^2 = 9.46$, $p < .002$ (see Table 2).

3.4 Detecting effective activity use

The findings of the previous section suggest that not all activity completion is active learning, and some might reflect "gaming the system" behaviors. This raises the important question of being able to distinguish effective active learning in activity use from other uses. From the previous analyses, we concluded spending more time, being more accurate across all attempts and achieving highest score with less attempts all predict better quiz performance and provide better fit to the data. Using these findings, we created two classifiers of "gaming the system" behaviors. One that takes only attempt duration into account and another that takes into account not only duration but also accuracy of each attempt.

Table 2. Summary of regression models used to evaluate the benefits of doing and reading in the online course.

Model	Number activities	eText pages	Added predictor	AIC	BIC
Baseline (before quiz start)	-0.04*	0.14***	-	9404.7	9442.2
Baseline (after quiz start)	0.18***	0.05**	-	9311.3	9348.8
Time working on activities	-0.05**	0.12***	0.10***	9377.4	9421.1
Average performance on activities	-0.02	0.14***	0.04**	9400.1	9443.8
Number-weighted best performance on activities	-0.03	0.14***	0.04**	9397.3	9441.0
Effective active learning activity use (duration-based)	0.29	0.14***	-0.33	9404.0	9447.8
Effective active learning activity use (duration+accuracy)	-0.21**	0.14***	0.17*	9400.1	9443.8

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

3.4.1 Duration-based classifier

Using the raw attempt log for each activity for each student, we determined whether each attempt was faster than what is "normal" for that student by considering that students' median time completing similar activities. An attempt was considered as not effective active engagement if it was shorter than the median of all attempts for that student for the same activity. Thus, in essence, this classifier positions each attempt as too quick to be likely to involve active engagement, based on how long students often take to complete similar activities, and is consistent with our findings in the previous section that time is a better predictor of effective activity use. This classifier identified approximately 47% of attempts before the quiz as not involving active engagement.

3.4.2 Duration+accuracy classifier

Using the raw attempt log for each activity for each student, we determined whether each attempt was faster and less accurate than what is "normal" for that student by considering that students' median time completing all activities and their median accuracy. The previous analyses suggested that accuracy in the activities was also a good predictor of effective activity use. Thus, the premise for this classifier was that if students were merely completing the same activity multiple times by randomly varying their answers until reaching high scores, one would expect that it would involve multiple short attempts with low accuracy. Active engagement, on the other hand, would involve longer attempts with higher accuracy. Approximately 22% of attempts were identified as fast and inaccurate by this classifier and were classified as non-effective activity use. These attempts were a subset of the attempts identified by the previous classifier, i.e., the low accuracy subset.

3.5 Effective activity use models

We included the counts of "effective activity use" from each classifier in two different models and compared the models with the baseline model. These analyses tell us whether, when considering effective activity use, we are able to capture the learning benefit of engaging with activities.

3.5.1 Active learning use of activities as identified by the duration-based classifier

When using the duration-based classifier, we found that the number of effective activity use was not related to quiz performance, β = -0.33, p = .101 and this model did not improve fit to the data, χ^2= 2.69, p =.103 (see Table 2).

3.5.2 Active learning use of activities as identified by the duration+accuracy classifier

When we considered the counts obtained using the duration+accuracy classifier, we found that greater effective activity use predicted better quiz performance, β= 0.17, t (3053) = 2.56, p =.011, and this effect was 1.2 times larger than that of accessing more eText pages, β= 0.14. This model provided a better fit to the data, χ^2= 6.63, p =.010 (see Table 2).

4. DISCUSSION

The two main aims of this study were (1) to investigate the relative benefits of completing activities versus reading in an online course in which completing activities was mandatory, and (2) to explore the key features of effective active engagement with activities and how to detect them in student online behavior.

Previous research suggests that the most beneficial practice activities involve effortful, active engagement and knowledge manipulation by the students [9,18,19]. Indeed, we found evidence that features connected with effort and engagement with the activities were better predictors of learning than completing activities per se (time spent and accuracy). However, overall, we found that, when activities are required and graded, completing more activities is not necessarily a good predictor of improved learning. Instead, spending more time completing the activities and being more accurate across attempts, are better predictors of improved quiz performance. These analyses offer the perfect case-study for the "doer effect" and the characteristics of the learning activities that contribute to improved learning outcomes.

Across all models, more reading (accessing more eText pages) remained the best overall predictor of learning outcomes, even when compared to features indicative of active engagement with the activities. There are multiple reasons for this finding. It is possible that students who accessed the eText were engaging in active learning by autonomously searching answers to activities. Indeed, in a departure from previous studies [13], the activities in this course offered only corrective feedback, implicitly encouraging students to seek more information in the eText, which might have contributed to the results presented here. Another possibility is that better students, who ultimately perform better in the course, access a course material that is not mandatory or rewarded. The reduced correlation between pages read after starting the quiz and quiz performance, suggest that this possibility of a third variable explanation is somewhat less likely than the first possibility that reading behavior in this course is associated with active learning of completing the activities because of the type of feedback used in the activities.

Another main novelty of the present work is the development of analytical processes to identify which activity engagement might be productive and which might not. Under the assumption that the same activity might be completed effortfully and involve knowledge manipulation or only involve "action", we developed two classifiers. The first classifier took into account only the duration of the attempt, whereas the second classifier took into account the duration as well as accuracy of the attempt. The outcome of the first detector did not seem to improve the model fits predicting quiz performance. Conversely when we tested activity use considering only attempts that were classified as effective active learning by the second classifier, we saw that greater effective activity use was a positive predictor of better quiz performance. In fact, greater effective activity use as defined by the second classifier resulted in 1.2 times better quiz performance than accessing more eText pages. Conversely, considering every activity attempt was a negative predictor of quiz performance.

The difference in outcomes between the two classifiers suggests that time to solve a problem by itself might not be sufficient to identify gaming. Fast but accurate attempts might be effective or at least do not negatively impact performance. One possibility is that students learn from fast correct attempts or that fast and correct attempts reflect already learned knowledge. This finding is also congruent with previous findings that some students or some activities might not be harmed by gaming [1,17].

Our approach to defining classifiers differs from previous approaches in educational data mining. We used an explanatory approach; our gaming classifiers were very simple and identified gaming events based on initial data analytics and the literature. This approach might yield less predictive models than previous efforts using more complex (and potentially more predictive) models [1]. However, one benefit of our approach is its explanatory power. The gaming detectors we created can not only identify gaming behavior from the student data but also contribute to a better understanding of what characterizes these types of behaviors (see also [17]).

The findings presented here are a critical first step towards developing effective active learning activities in online courses. Given the greater student autonomy often associated with online courses, it is important to develop methods to identify effective activity use. Critical next steps are to create generalized detectors that can be used online to provide students with feedback not only about the content of the activity, but also their use of the activity as active learning tool. For example, the activity could alert the student to the fast pace and low accuracy and suggest that they try a different approach to the task. Similar classifiers of these "gaming the system" behaviors have been suggested before in the context of intelligent tutoring systems with good success [3].

In sum, the work presented here suggests that not all activity use is active learning and therefore contributes to better learning outcomes. Some activity use might reflect "gaming the system" behaviors that might yield high immediate scores but are not reflective of better learning and later quiz performance (for a discussion see [5]). Similarly, not all reading is passive learning, and intentional use of reading materials might reflect active learning. Accordingly, it is important to be able to detect when students are engaging in active learning and when they are not, regardless of the type of learning activity. The current work establishes an initial step in that direction by identifying which features are associated with active learning engagement when students' complete activities in online courses, and by developing classifiers of this type of behavior that can be adapted and generalized to other courses.

5. ACKNOWLEDGMENTS

This work was supported in part by a National Science Foundation grant (ACI-1443068) toward the creation of LearnSphere.org and by funding from Google to K.R.K.

6. REFERENCES

1. Ryan S. J. d. Baker, Albert T. Corbett, Ido Roll, and Kenneth R. Koedinger. 2008. Developing a generalizable detector of when students game the system. *User Modeling and User-Adapted Interaction* 18, 3: 287–314.
2. Ryan SJ d Baker, Albert T. Corbett, Kenneth R. Koedinger, Shelley Evenson, Ido Roll, Angela Z. Wagner, Meghan Naim, Jay Raspat, Daniel J. Baker, and Joseph E. Beck. 2006. Adapting to when students game an intelligent tutoring system. In *International Conference on Intelligent Tutoring Systems*, 392–401.
3. Ryan Baker, Jason Walonoski, Neil Heffernan, Ido Roll, Albert Corbett, and Kenneth Koedinger. 2008. Why Students Engage in "Gaming the System" Behavior in Interactive Learning Environments. *Journal of Interactive Learning Research; Charlottesville* 19, 2: 185–224.
4. Lucy Barnard, William Y. Lan, Yen M. To, Valerie Osland Paton, and Shu-Ling Lai. 2009. Measuring self-regulation in online and blended learning environments. *The Internet and Higher Education* 12, 1: 1–6. https://doi.org/10/b6sf5z
5. Robert A. Bjork, John Dunlosky, and Nate Kornell. 2013. Self-Regulated Learning: Beliefs, Techniques, and Illusions. *Annual Review of Psychology* 64, 1: 417–444.
6. Paulo F. Carvalho, David W. Braithwaite, Joshua R. de Leeuw, Benjamin A. Motz, and Robert L. Goldstone. 2016. An In Vivo Study of Self-Regulated Study Sequencing in Introductory Psychology Courses. *PLOS ONE* 11, 3: e0152115–e0152115.
7. Paulo F. Carvalho, Elizabeth A. McLaughlin, and Kenneth R. Koedinger. 2017. Is there an explicit learning bias? Students beliefs, behaviors and learning outcomes. In *Proceedings of the 39th Annual Conference of the Cognitive Science Society*, 204–209.
8. Donald S. Ciccone and John W. Brelsford. 1976. Spacing repetitions in paired-associate learning: Experimenter versus subject control. *Journal of Experimental Psychology: Human Learning and Memory* 2, 4: 446–455.
9. L. Deslauriers, E. Schelew, and C. Wieman. 2011. Improved Learning in a Large-Enrollment Physics Class. *Science* 332, 6031: 862–864. https://doi.org/10.1126/science.1201783
10. Jonathan Fernandez and Eric Jamet. 2017. Extending the testing effect to self-regulated learning. *Metacognition and Learning* 12, 2: 131–156. https://doi.org/10/gbms77
11. S. Freeman, S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth. 2014. Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences* 111, 23: 8410–8415.
12. René F. Kizilcec and Geoffrey L. Cohen. 2017. Eight-minute self-regulation intervention raises educational attainment at scale in individualist but not collectivist cultures. *Proceedings of the National Academy of Sciences* 114, 17: 4348–4353.
13. Kenneth R. Koedinger, Jihee Kim, Julianna Zhuxin Jia, Elizabeth A. McLaughlin, and Norman L. Bier. 2015. Learning is not a spectator sport: doing is better than watching for learning from a MOOC. *Proceedings of the Second (2015) ACM Conference on Learning @ Scale - L@S '15*: 111–120.
14. Kenneth R. Koedinger, Elizabeth A. McLaughlin, Julianna Zhuxin Jia, and Norman L. Bier. 2016. Is the doer effect a causal relationship? *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge - LAK '16*: 388–397. https://doi.org/10.1145/2883851.2883957
15. Norbert Michel, J J Cater, and Otmar Varela. 2009. Active versus passive teaching styles: An empirical study of student learning outcomes. *Human Resource Development ...* 20, 4: 397–418. https://doi.org/10.1002/hrdq
16. Kayla Morehead, Matthew G Rhodes, and Sarah DeLozier. 2016. Instructor and student knowledge of study strategies. *Memory* 24, 2: 257–271.
17. Kasia Muldner, Winslow Burleson, Brett Van de Sande, and Kurt VanLehn. 2011. An analysis of students' gaming behaviors in an intelligent tutoring system: predictors and impacts. *User Modeling and User-Adapted Interaction* 21, 1–2: 99–135. https://doi.org/10/bvkpzd
18. Henry L. Roediger and Jeffrey D. Karpicke. 2006. The Power of Testing Memory: Basic Research and Implications for Educational Practice. *Perspectives on Psychological Science* 1, 3: 181–210.
19. Carl E Wieman. 2014. Large-scale comparison of science teaching methods sends clear message. *Proceedings of the National Academy of Sciences of the United States of America* 111, 23: 8319–8320.

Finding Topics in Enrollment Data

Benjamin Motz, Thomas Busey, Martin Rickert, and David Landy
Department of Psychological and Brain Sciences
Indiana University, Bloomington, Indiana
bmotz@indiana.edu, busey@indiana.edu, rickertm@indiana.edu, dlandy@indiana.edu

ABSTRACT

Analyses of student data in post-secondary education should be sensitive to the fact that there are many different topics of study. These different areas will interest different kinds of students, and entail different experiences and learning activities. However, it can be challenging to identify the distinct academic themes that students might pursue in higher education, where students commonly have the freedom to sample from thousands of courses in dozens of degree programs. In this paper, we describe the use of topic modeling to identify distinct themes of study and classify students according their observed course enrollments, and present possible applications of this technique for the broader field of educational data mining.

Keywords

Higher education, Learning analytics, Topic modeling, Student data, Course enrollment, Transcripts, Educational data mining

1. INTRODUCTION

At any large educational institution, the student population is likely to be rather heterogeneous. One prominent source of variability is the range of academic topics available to students, reflected in the breadth of courses available to them, and the diverse requirements of numerous degree and pre-professional programs. This variability can make it challenging to analytically characterize the behaviors of students (e.g., graduation rates, engagement, grades), because students with different academic interests will have different experiences of higher education.

But nevertheless, some students will share similar experiences. There may be ways of parceling the diverse population to identify distinct groups of students whose academic interests are relatively homogenous within-groups, but differ between-groups. A simplistic strategy would divide students by major; but it may be desirable to identify groups before students have explicitly declared a degree program. In addition, these data may be unreliable as students often switch majors, and majors may artificially segregate students with generally similar interests and behaviors (e.g., students majoring in Chemistry, Biochemistry, or Biotechnology are probably quite similar). And dividing students by major may also be circular: Do majors describe student interests or merely describe the administrative landscape of degree programs? Instead of segmenting by major, the analytical challenge is to identify distinct areas of study directly from student course enrollments, where students assigned to each area have similar academic interests, experiences, and behaviors.

In educational data mining, clustering is the most commonly-applied method for classifying students [3, 19]. Vellido et al. [21] recently summarized a range of cluster analysis techniques, reviewed their applications to educational data mining, and compiled a bibliography of published studies that pursued such applications, particularly in e-learning environments.

It is conceivable that one might perform cluster analysis with course enrollments, as recorded on student transcripts. Each individual course might be treated as a single dimension in a high-dimensional space (e.g., one dimension for every course), and a transcript would be a single point in this space (with enrollments, 0 or 1, along each dimension). But there are major problems with this approach, particularly the "curse of dimensionality." In high-dimensional space, the data become sparse, and distances between individual points become almost equal, often yielding meaningless clustering results [1]. Recently-developed algorithms and distance metrics may improve the performance of high-dimensional clustering [13], but in this paper, we propose an entirely different approach that is better-suited to this particular analytical case.

We propose the use of topic modeling to address the challenge of classifying student transcripts. Topic modeling is commonly used for natural language processing applications (e.g., [10]) to identify abstract themes, or "topics," that exist in a collection of documents by analyzing the statistical distribution of words across these documents (for a review, see [5]). For our purposes, each document is a student transcript and each word is a course enrollment.

2. METHOD AND CONSIDERATIONS

Topic modeling is an umbrella term for a handful of methods that accomplish similar goals. The most popular method, and the one that we recommend for this application, is Latent Dirichlet allocation (LDA; [6]). Intuitively, in its simplest form, the approach initializes by assigning every token (each word in each document, or in the present analysis, each course in each transcript) to a random topic, and then repetitively iterates through the tokens, updating topic assignments in order to reduce the occurrence of individual words across multiple topics, while still preserving the contexts of words that tend to appear together within individual documents. Ultimately the method will produce a model of topics, a description of the words that tend to occur together.

Topic modeling has many advantages for the purposes of classifying academic topics:

- Rather than unequivocally classifying documents to topics, LDA assigns each word to a topic, producing a distribution of topic assignments for each document, and a probabilistic distribution of words for each topic (similar to soft-clustering approaches [18]). For educational data mining purposes, this is advantageous because a single course might occur in several topics with different probabilities, depending on the course's context in different students' transcripts (e.g., the course "Elementary Calculus" might be differentially-predictive for students interested in Biology vs. Computer Science). In order to produce a coarse classification of students, we can simply assign students to the topic that appears most frequently in their transcripts.

- LDA is insensitive to the order of words (although it needn't be; [22]), which therefore allows the analysis of courses that are not taken in a strict sequence
- LDA is also generally insensitive to the length of documents (for documents with a small number of words, the prior probability of the topic distribution across all documents has a larger effect), allowing the analysis of incomplete transcripts.
- LDA can be parameterized in a way that tends to yield similarly-sized topics, minimizing the possibility of disproportionately large or small groups (which tends to occur with clustering).

Many software implementations of topic modeling methods are available[1]; we selected MALLET [15], which is open-source and has a large community of active users.

2.1 Case Data and Preprocessing

We identified all full-time students enrolled in a baccalaureate program at Indiana University Bloomington who initially became new or transfer students between 1995 and 2009. Students who did not complete any courses at our local institution were excluded. We then constructed course identifiers (which served as "words") by concatenating the academic program code, the course inventory, and the course number for every enrolled course appearing on the students' transcripts[2], irrespective of earned grade. There were 9,566 unique courses, 86,808 unique students, and students had an average of 29.3 courses listed on each transcript.

2.2 Modeling Topics

In traditional lexical analyses, documents contain words, and the topic model probabilistically associates the latent topics to each document through the words that it contains. In our analysis, courses were treated as words, and each student was represented by a document, the student transcript, that contained a collection of all courses taken by the student as part of their undergraduate education. Thus we are able to associate both students and courses with the discovered latent topics.

In its most basic form, the only parameter that needs to be supplied when modeling topics is the desired number of topics (see Section 2.3, below).

While there are various ways to visualize extracted topics (e.g., [8]), perhaps the easiest way to summarize a topic model is to present the words that are most probable in a particular topic for a set of representative topics, sometimes called "topic keys." A summary of a topic model on our transcript dataset, describing 6 of 24 topics, is shown in Figure 1. An interpretive gloss (in quotations) is provided above the ten most probable courses for that topic, listed in descending order (labels for the full set of 24 topics are shown on the right side of Figure 3, which is described later in this article). Students were assigned to the topic that appeared most frequently on their transcript, and the percent of the full student cohort that was assigned to each topic is also provided next to the topic label (if students had been evenly-allocated to the 24 topics, there would be 4.2% of the cohort in each topic). At face value, the algorithm did an impressive job of allocating the nearly 10,000 courses into distinct academic topics, particularly when considering that the model is entirely unsupervised. These topics were identified simply by analyzing the contextual trends in students' transcripts.

Importantly, one should not assume that these topics would emerge if the same analysis were performed on a different dataset. Different institutions have different academic programs and requirements, and different enrollment patterns. The current results are presented as a methodological case study, not as results that should be expected to generalize.

Some predictable patterns emerge in the current dataset, such as topics that clearly reflect the curriculum of popular majors, including "Business" and "Psychology." Other topics seem to slice across traditional academic silos, such as "Language Education," or the "Government" topic, which features courses from the Department of History and also from the Department of Political Science, even though neither department's undergraduate degree program explicitly requires courses from the other. Yet other topics seem to identify subgroups within a field, such as "Health Science" and "Basic Science," which segregates premedical interests from more basic science coursework, even though many of the students assigned to these topics are pursuing the same undergraduate degrees (e.g., Biology).

An essential caveat with topic modeling is that the algorithm yields a description of latent topics (in this case, themes of undergraduate study), but does not describe the behavior of any individual student. The topics can be used to partition students into distinct groups (e.g., by assigning a student to the most frequent topic in their transcript), but the topics themselves do not characterize individual students with any specificity. Rather, they describe statistically separable academic themes. When interpreting topic models, it is important to remember that the topics characterize themes of study, but individual student behaviors may be more complex, as any student's transcript would be expected to contain courses from multiple topics with different frequencies.

2.3 The Number of Topics

Topic modeling requires that the analyst specify the appropriate number of topics (T) in the dataset. For some applications, T may be a known quantity; perhaps there are predetermined academic tracks that any student might pursue, and the goal is simply to characterize the enrollments that co-occur with these known topics. However, for most analyses, the number of topics is unknown, and the analyst must determine the appropriate number of topics to account for information in the dataset, according to the desired granularity of the analysis. There are methods for automatically inferring optimal values for T according to model performance measures [2, 16], but we preferred a more exploratory approach. Specifically, we extracted topics for a range of desirable values for T, evaluated these models using hold-out data, and then selected a value T to maximize likelihood while

[1] David Mimno maintains a reference list including software tools for topic modeling: http://mimno.infosci.cornell.edu/topics.html

[2] When dealing with this type of codified data, with course identifiers that may include numbers and punctuation symbols, it is important to specify the structure of the words as a regular expression in the analysis software, so that the documents are parsed appropriately.

"GOVERNMENT" (3.2% of cohort)	*"BUSINESS"* (12.5% of cohort)	*"PSYCHOLOGY"* (4.6% of cohort)
POLSY200: Contemporary Political Topics	**BUSX201**: Technology & Business Analysis	**PSYP324**: Abnormal Psychology
POLSY103: Introduction to American Politics	**BUSX420**: Business Career Planning	**PSYK300**: Statistical Techniques
HISTH105: American History I	**BUSA202**: Intro to Managerial Accounting	**PSYP102**: Introductory Psychology II
HISTA300: Issues in United States History	**BUSX220**: Career Perspectives	**PSYP199**: Career Planning for Psychology
COASW333: Intensive Writing	**BUSZ302**: Managing & Behavior in Organizations	**PSYP151**: Introductory Psychology I for Majors
POLSY109: Introduction to International Relations	**BUSF370**: Integrated Business - Finance	**PSYP335**: Cognitive Psychology
HISTB300: Issues in Western European History	**ECONE370**: Statistical Analysis for Business	**PSYP320**: Social Psychology
HISTH106: American History II	**BUSX204**: Business Communication	**PSYP211**: Methods in Experimental Psychology
POLSY100: American Political Controversies	**BUSP370**: Integrated Business - Operations	**PSYP315**: Developmental Psychology
HISTJ300: Seminar in History	**BUSJ370**: Integrated Business - Strategy	**PSYP152**: Introductory Psychology II for Majors

"LANGUAGE EDUCATION" (4.1% of cohort)	*"HEALTH SCIENCE"* (4% of cohort)	*"BASIC SCIENCE"* (7.8% of cohort)
HISPS275: Intro to Hispanic Culture	**ANATA215**: Basic Human Anatomy	**CHEMC117**: Principles of Chemistry II
HISPS310: Intro to Hispanic Linguistics	**MSCIM131**: Disease and the Human Body	**BIOLL112**: Biological Mechanisms
COASW333: Intensive Writing	**SOCS100**: Introduction to Sociology	**BIOLL113**: Biology Laboratory
ENGL202: Literary Interpretation	**PSYP101**: Introductory Psychology I	**BIOLL111**: Evolution & Diversity
EDUCM300: Teaching in Pluralistic Society	**PHSLP215**: Basic Human Physiology	**PHYSP201**: General Physics I
ENGW203: Creative Writing	**CHEMC101**: Elementary Chemistry	**CHEMC341**: Organic Chemistry I
HISPS331: The Hispanic World	**ENGW131**: Elementary Composition	**BIOLL211**: Molecular Biology
ENGW103: Introductory Creative Writing	**PSYP102**: Introductory Psychology II	**PHYSP202**: General Physics II
HISPS317: Spanish Conversation & Diction	**CLASC209**: Medical Terms from Greek & Latin	**CHEMC105**: Principles of Chemistry I
EDUCH340: Education & American Culture	**HPERH160**: First Aid and Emergency Care	**CHEMC342**: Organic Chemistry II

Figure 1: Top 10 most probable courses for 6 representative topics (of 24 total). These results are provided for illustration purposes, and topics will likely vary between institutions.

balancing the risk of overfitting with too many topics for the intended analysis.

2.3.1 Evaluating using hold-out data

Current guidelines for topic model evaluation were proposed by Wallach et al. [23]. This approach seeks to estimate the probability of hold-out data, given a particular topic model. The first step was to segregate the documents into a training set (random 90% of documents) and a hold-out set (remaining 10%). Topic models were then developed on the training set for a range of reasonable values for T (we used 2 to 100, in steps of 2). For each of these models, we estimated a log likelihood (LL) value for every document in the hold-out set, given that particular model. LL is a negative number, and intuitively, it provides an estimate of how unexpected the hold-out document's collection of words would be, considering the model's configuration of topics; LL values closer to zero indicate that the model was better, as any given document was less unexpected. Because the evaluation process is non-deterministic, we repeated the evaluation process 10 times, averaged the LL for each document across these 10 runs to obtain a more stable probability estimate, and then summed the averaged LL across all hold-out documents. This summed, averaged LL was finally divided by the total number of tokens in the hold-out set to produce a normalized-LL estimate; in many studies, this normalized value ranges between -10 and -6. The solid black line in Figure 2A illustrates the averaged LL/token for topic models on student transcripts, for a range of T.

2.3.2 Finding the inflection point

Ultimately, the desired number of topics should be determined through a combination of statistical analysis, general insights into the structure of the data, and consideration of the purpose of the model. Increasing the number of topics will generally improve the LL/token estimates, but above a certain point, these incremental improvements are trivial. For the current application, we sought to determine the fewest number of topics, such that additional topics would yield minimal improvements to the quality of the model. To find this inflection point, we fit a piecewise linear regression model on the LL/token estimates, seeking the value T* that minimized the root mean square error of the linear trends, T<T* and T>T*. As illustrated by the dashed lines in Figure 2A, this point was T*=24 topics; importantly, the topic keys (six are shown in Figure 1) made intuitive sense, and yielded an insightful model for this analysis.

2.3.3 Stop lists and frequent courses

There is a convention in topic modeling to remove high-frequency tokens from the training dataset. When modeling topics in linguistic corpora, this pre-processing step is intended to filter words that do not contain meaning (such as “the”, “a”, “of”, etc.) and would add unnecessary noise to the identification of topics. These excluded tokens are called a stop list.

Along these lines, it may be useful to exclude high frequency courses when modeling academic topics. Courses that appear on a large proportion of student transcripts (general education courses, high-enrollment prerequisites, etc.) may be practically meaningless for the purposes of classifying student interests. However, there has been relatively little empirical work evaluating the use of stop lists when modeling topics. In information retrieval algorithms more generally, Manning et al. [14] note that the cost of including high-frequency tokens (in computational time) is minimal, and that the recent trend is to use smaller stop lists, if any at all.

We approached this issue as an empirical question (i.e., a sensitivity analysis): Will the use of a stop list affect model

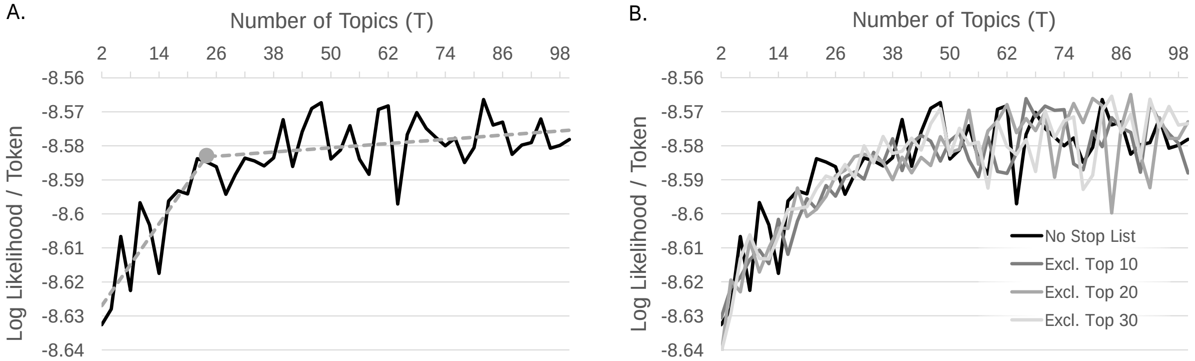


Figure 2: Model performance for a range of values for T (A), and comparing stop lists (B).

performance? The previously-described modeling analysis was repeated three additional times, filtering the top-10, top-20, and top-30 most-frequent courses (the 10th-most-frequent course appears in 19.7% of transcripts, the 20th- appears in 13.0%, and the 30th- appears in 10.5%). For reference, the highest-frequency course at our institution (Elementary Composition) appeared on 46.3% of student transcripts. The log-likelihood of these models, with T ranging from 2-100, is illustrated in Figure 2B.

There was no clear effect of including a stop list on the model's performance, irrespective of the number of topics or the number of words on the stop list. It may be that, at our institution, the highest-enrollment courses impart a minor amount of information about the thematic structure of a student's enrollments, but not enough to substantively improve or impair the model's performance. For most natural language processing applications, stop lists typically aim to filter words that occur in a very large proportion (e.g. 85%) of documents, so even though they're relatively popular at an institutional level, the highest-enrollment courses (appearing in less than 50% of transcripts) simply might not rise to the level of frequency that would merit their exclusion. These effects may vary across institutions, but without a clear separation in model performance, we suggest including all relevant courses in the analysis, and do not advocate the use of a stop list.

3. EXAMPLE APPLICATIONS

In a rapidly growing field such as educational data mining, it is difficult to anticipate the full range of uses of a relatively new method, such as topic modeling, or any analytical technique. The following three examples are only intended to help illustrate, at a very high level, the general value of identifying academic topics, and the wide range of potential applications.

3.1 Sandwich Estimator

In educational data mining, researchers commonly try to predict the effect of one variable on another variable, such as the effect of an automated flagging system on graduation rates. Common modeling approaches (such as ordinary least squares regression) typically carry the assumption that each observation is independent from the others. But in higher education, this is a weak assumption. Different students are jointly exposed to the same classes, instructors, student groups, and graduation requirements, and moreover, they might be expected to communicate with each other about these experiences, and influence each other's behaviors. Although violating the independence assumption will not affect the point estimate (i.e., magnitude) of a regression parameter, it can significantly change the interval estimate (i.e., precision) of the parameter, which in turn, changes the probability of making a Type I or Type II error.

One solution to this issue would be to fit multilevel random effects models to account for the non-independence of observations and the cross-classified data structure (with students not strictly nested within grouping variables). However, this would be an absurdly complex model, with every course, semester, instructor, etc. included as a crossed random effect; we feel that such an effort is impractical.

But considering that topic models are derived from patterns in course enrollments, the topic classifications can be used as a grouping variable that will account for the non-independence of student experiences and produce corrected (i.e., sandwich) estimates of the standard errors for the model parameters [24]. By classifying students according to the most frequent topic in their transcript, we are able to identify subgroups of students such that their coursework and learning activities are correlated within-groups, and are independent between-groups. In our enrollment data, using T=24 and a binary response variable indexing graduation within four years of initial enrollment, we obtained an estimated intraclass correlation of 0.254. This suggests that about a quarter of the variance of within-class 4-year graduation rates are explained by topic assignment, heteroscedasticity that can be easily corrected in regression models.

3.2 The Alignment of Programs and Topics

There are latent interests held by students that influence the courses they select. Sometimes these enrollment choices are codified in degree requirements or prerequisites, or even by external forces (such as medical school requirements). However, as mentioned in the Introduction, the nuanced boundaries that delineate different degrees do not necessarily provide a fair representation of the different topical interests that might motivate students' course selections. This relative alignment of degree programs with students' interests can be investigated using topic modeling.

For example, in discussions of such academic restructuring, it is often suggested that departments with similar interests should merge or combine resources [11]. The current topic modeling approach may reveal different degree programs that are jointly represented by a single topic, and these might be candidates for this type of restructuring. At our institution, our analysis reveals notable overlap between History and Political Science, and there may be administrative synergies between these programs.

In contrast, there may be topics that integrate courses from different departments in stable ways that are unaccommodated by any degree. Beyond mere overlap between programs, students may be sampling courses from multiple programs to construct "hidden majors," academic chimeras that may not exist as formalized degree programs, but that integrate diverse coursework to create stable topics of interest. For example, course enrollments at our institution revealed a "Media Studies" topic of study that was not accommodated by any single major; it blended courses from Communications, Comparative Literature, Sociology, and more. Our discovery of this topic provided support for our institution's recent initiative to create a new Media School.

And topic modeling might also be used to reveal separable sub-disciplines within a single degree program. Even within an individual major (such as Psychology) there is ample opportunity for students to focus on subdisciplines (such as counseling, human factors, child development, behavioral research, etc.). Just as topic modeling can reveal latent academic themes in an entire university's course catalogue, it can also be applied to a single academic division or program, to evaluate the thematic structure within a single unit. At our institution, by modeling topics from the enrollments of recent graduates in Psychology, we identified themes related to law, medicine, and social psychology. These have enabled us to tailor career planning events, course offerings, and advising materials to the specific interests of our students.

3.3 Transitions and Outcomes

At colleges with flexible degree requirements, undergraduate students typically undergo an academic metamorphosis, enrolling in first-year general survey courses to eventually enrolling in specialized advanced courses [4]. This transition, from the nonspecific enrollment behaviors of freshmen to the niche upper-division coursework of soon-to-be graduates, is an area that has begun to receive increasing attention in higher education research, particularly in efforts to improve retention and eliminate boundaries and bottlenecks to STEM fields. By characterizing the various transitional paths from first year study to subsequent disciplinary specialization (and the success rates associated with these paths), institutions would be better-equipped to test hypotheses about pipeline issues, and to develop effective advising strategies and interventions for beginning students [12].

Along these lines, topic modeling might be applied to first year enrollments, in order to identify the broad thematic enrollment trends of beginning students. And then we might draw the paths from first year topics to the topics derived from full transcripts, to illustrate how students transition from initial coursework to eventual specialization in an established topic. This analysis is described below, and illustrated in Figure 3.

3.3.1 Visualizing Paths from First Year Topics

The previously-described topic modeling approach was performed on the same set of students, but we limited their transcripts to only include courses that were credited during the student's first year of study. There were 3,330 distinct courses on these truncated transcripts, and on average, there were 9.0 courses per student during this first year. After evaluating models for a range of values for T we found an inflection point at 5 topics, and determined that this provided the appropriate balance of model

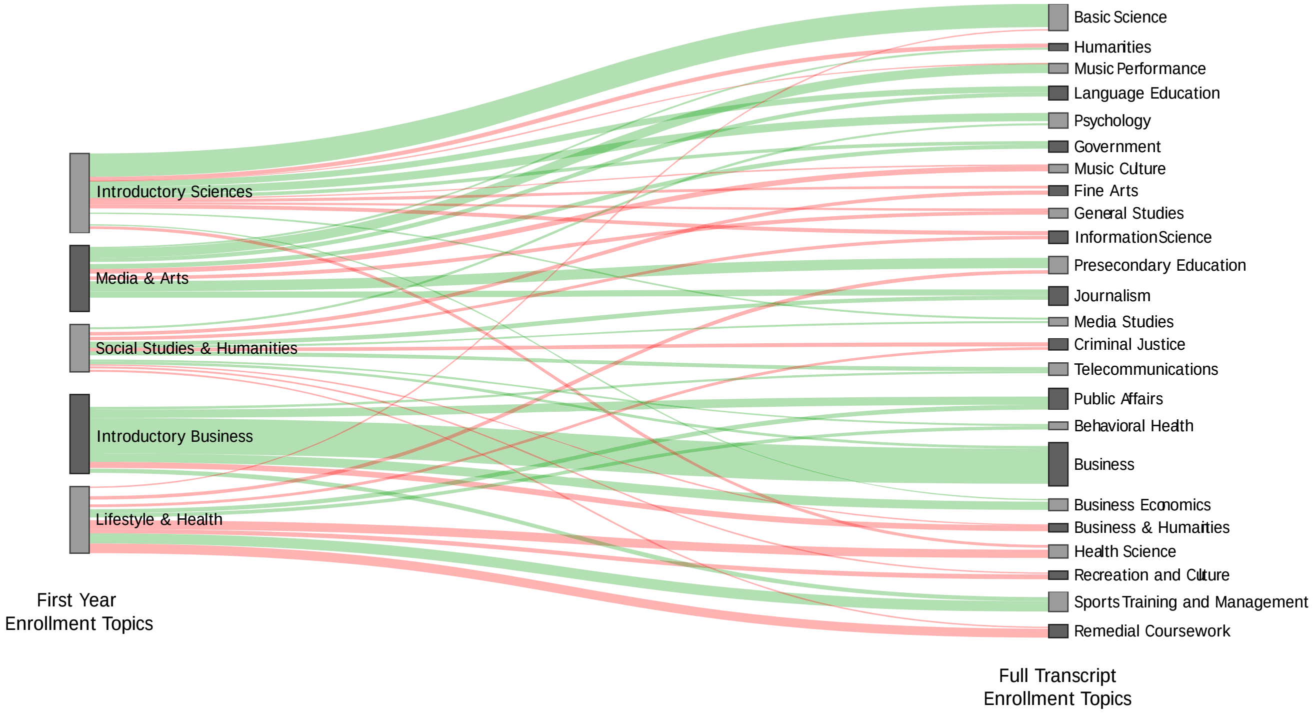


Figure 3: Diagram of transitions between first year and full transcript enrollment topics.

performance and face validity. And, as previously discussed, students were assigned to the topic that appeared most frequently on their first-year transcripts. For every student in our sample, we now had one topic assignment (1-5) for their first year enrollments, and another topic assignment (1-24) for their complete transcript.

For each student, we also identified whether they received a baccalaureate degree up to 4 years after their initial enrollment. For this sample of students during this time frame, the overall 4-year graduation rate was 51.2%. Of course, many more students will eventually receive a degree after their 4th year, but this 4-year rate is relevant for institutional benchmarking purposes.

Figure 3 was produced using the Sankey diagram plugin for D3 [7]. Paths are green or red if the students within that path have a 4-year graduation rate above or below 51.2%, respectively. The relative size of the gray boxes and colored paths represent the number of students assigned to the topic or transition. To make this diagram more readable, we have only included the two largest entry paths for each of the full-transcript enrollment topics.

3.3.2 Interpreting Topic Transitions

One of the immediate observations from this analysis is that a student's first year enrollments tend to be reasonably predictive of the themes of study where the student may ultimately arrive at the end of their career—the flat paths tend to be thicker than the sloped cross-cutting paths. Initially one might attribute this to the fact that first-year enrollments are included in the full-enrollment transcripts. This artifact may play a role, however, looking back to Figure 1, an important observation is that the most probable courses for full-enrollment topics (those at the top of the list) are commonly 200-level courses, typically beyond the first year (100-level) introductory sequence. We observe that students' first year enrollments are not dissociated from their future enrollment tendencies.

This observation might suggest that students who transition to a relatively unrelated topic after their first year would be at a disadvantage to graduate in 4 years. But the data seem to suggest otherwise: that some full-transcript topics simply have lower 4-year graduation rates than others, regardless of whether the students followed a straight thematic trajectory, or seemed to originate from an untraditional first-year topic. For example, students who ultimately study "Recreation and Culture" have lower graduation rates, regardless of whether they began college by studying "Lifestyle and Health" (a structurally similar theme) or "Social Studies and Humanities" (a relatively distant theme).

These exploratory analyses and interpretations have their limitations, and the hypotheses derived from a visualization like this should receive further scrutiny on the local level. As discussed previously, our topic models describe abstract themes of study, and do not characterize students per se. The students whom we've identified as being members of a theme (because the theme appears most commonly on their transcript) may have other similarities, besides their course enrollments (e.g., third variables such as family expectations, cultural values), that contribute to their graduation rates or enrollment behaviors more directly than their coursework. Nevertheless, being able to easily visualize the flow of the entire student body (albeit indirectly) across the academic landscape can serve useful purposes toward understanding the inflow into a particular area, and ultimately developing better-informed advising strategies.

4. CONCLUSION

Blanket generalizations that treat an institution's "students" as a single group are likely to be either ineffectively vague, or not applicable to all members of the student population [20]. In the classroom, post-secondary instructors find value in knowing the differentiating characteristics of the students in their classes, and tailoring instruction to accommodate their unique attributes [17]. Data-driven interventions and analytical characterizations of student behaviors should also be sensitive to the differences between students. In this paper, we've described an effective method for identifying one prominent source of variability: students' academic interests. By applying topic modeling to student transcripts, we are able to identify separable topics of study at our institution, and these topics can be further used to roughly classify students into distinct groups that feature similar enrollment behaviors.

Considering that it was originally developed as a natural language processing tool, topic modeling has well-documented applications to educational data mining in the analysis of student discourse (e.g., in a discussion forum; [9]) or written coursework, but it could also be applied to any form of unstructured categorical data at the university, such as LMS web traffic, library checkouts, or even meal point expenditures. Similarly, we believe that topic modeling is a straightforward and uniquely suitable method for identifying patterns in raw enrollment data.

5. ACKNOWLEDGMENTS

This work was supported by a grant from the Bay View Alliance. Special thanks to Mark Steyvers for his expertise and guidance, as well as Linda Shepard, Stefano Fiorini, and Mike Sauer for access to and assistance with the institutional data used in this analysis.

6. REFERENCES

[1] Aggarwal, C. C., Hinneburg, A., and Keim, D. A. 2001. On the surprising behavior of distance metrics in high dimensional spaces. *ICDT, Lect. Notes Comput. Sc.*, 1973 (Oct. 2001), 420-434. DOI= http://dx.doi.org/10.1007/3-540-44503-X_27

[2] Arun, R., Suresh, V., Madhavan, C. V., & Murthy, M. N. 2010. On finding the natural number of topics with Latent Dirichlet allocation: Some observations. In *Advances in Knowledge Discovery and Data Mining*, Springer, Berlin. 391-402.

[3] Baker, R. S. J. d. 2010. Data mining for education. In *International Encyclopedia of Education*, B. McGaw, P. Peterson, E. Baker, Eds. Elsevier, Oxford. 112-118.

[4] Babad, E., Darley, J. M., Kaplowitz, H. 1999. Developmental aspects in students' course selection. *J Educ. Psychol.*, 91, 1 (Mar. 1999), 157-168. DOI= http://dx.doi.org/10.1037/0022-0663.91.1.157

[5] Blei, D. M. 2012. Probabilistic topic models. *Commun. ACM*, 55, 4, 77-84. DOI= http://doi.acm.org/10.1145/2133806.2133826

[6] Blei, D. M., Ng, A. Y., and Jordan, M. I. 2003. Latent Dirichlet allocation. *J. Mach. Learn. Res.*, 3 (Jan. 2003), 993-1022.

[7] Bostock, M., Ogievetsky, V., and Heer, J. 2011. D^3 Data-Driven Documents. *IEEE T. Vis. Comput. Gr.*, 17, 12 (Dec. 2011), 2301-2309. DOI= http://dx.doi.org/10.1109/TVCG.2011.185

[8] Chaney, A. J. B. and Blei, D. M. 2012. Visualizing topic models. In *International AAAI Conference on Web and Social Media* (ICWSM '12). AAAI Press.

[9] Ezen-Can, A., Boyer, K. E., Kellogg, S., and Booth, S. 2015. 2015. Unsupervised modeling for understanding MOOC discussion forums: a learning analytics approach. In *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge* (LAK '15). ACM, New York, NY, USA, 146-150. DOI= http://dx.doi.org/10.1145/2723576.2723589

[10] Griffiths, T. L. and Steyvers, M. 2004. Finding scientific topics. *P. Natl. Acad. Sci. USA*, 101, suppl. 1 (Apr. 2004), 5228-5235. DOI= http://dx.doi.org/ 10.1073/pnas.0307752101

[11] Gumport, P. J. 2000. Academic restructuring: Organizational change and institutional imperatives. *High. Educ.*, 39, 1 (Jan. 2000), 67-91. DOI= http://dx.doi.org/ 10.1023/A:1003859026301

[12] Heileman, G. L., Babbitt, T. H., and Abdallah, C. T. 2015. Visualizing student flows: Busting myths about student movement and success. *Change: The Mag. of High. Educ.*, 47, 3 (Jun. 2015), 30-39. DOI= http://dx.doi.org/10.1080/00091383.2015.1031620

[13] Kriegel, H. P., Kröger, P., and Zimek, A. 2009. Clustering high-dimensional data. *ACM Trans. Knowl. Discov. Data*, 3, 1, Article 1 (Mar. 2009). DOI= http://dx.doi.org/10.1145/1497577.1497578

[14] Manning, C. D., Raghavan, P., and Schütze, H. 2008. *Introduction to information retrieval*. Cambridge University Press, Cambridge.

[15] McCallum, A. K. 2002. MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu

[16] Mimno, D. and Blei, D. 2011. Bayesian checking for topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (EMNLP '11). Association for Computational Linguistics, Stroudsburg, PA, USA, 227-237.

[17] Motz, B. A., Teague, J. A., Shepard, L. L. 2015. Know thy students: Providing aggregate student data to instructors. *EDUCAUSE Review Online* (Mar. 2015).

[18] Peters, G., Crespo, F., Lingras, P., and Weber, R. 2013. Soft clustering – Fuzzy and rough approaches and their extensions and derivatives. *Int. J. Approx. Reason.*, 24, 2, 307-322. DOI=https://doi.org/10.1016/j.ijar.2012.10.003

[19] Romero, C. and Ventura, S. 2010. Educational data mining: A review of the state-of-the-art. *IEEE T. Syst. Man Cy. C*, 40, 6 (Nov. 2010), 601-618. DOI= http://dx.doi.org/10.1109/TSMCC.2010.2053532

[20] Quaye, S. J. and Harper, S. R. 2014. *Student engagement in higher education: Theoretical perspectives and practical approaches for diverse populations*. Routledge, New York.

[21] Vellido, A., Castro, F., and Nebot, A. 2010. Clustering educational data. In *Handbook of Educational Data Mining*, C. Romero, S. Ventura, M. Pechenizkiy, R. S. J. d. Baker, Eds. CRC Press, Boca Raton, FL. 75-92.

[22] Wallach, H. M. 2006. Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning* (ICML '06). ACM, New York, NY, 977-984. DOI=http://dx.doi.org/10.1145/1143844.1143967

[23] Wallach, H. M., Murray, I., Salakhutdinov, R., and Mimno, D. 2009. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning* (ICML '09). ACM, New York, NY, 1105-1112. DOI= http://dx.doi.org/10.1145/1553374.1553515

[24] Williams, R. L. 2000. A note on robust variance estimation for cluster-correlated data. *Biometrics*, 56, 2 (Jun. 2000), 645-646. DOI= http://dx.doi.org/10.1111/j.0006-341X.2000.00645

Can Textbook Annotations Serve as an Early Predictor of Student Learning?

Adam Winchell
University of Colorado Boulder
adam.winchell@colorado.edu

Michael Mozer
University of Colorado Boulder
mozer@colorado.edu

Andrew Lan
Princeton University
andrew.lan@princeton.edu

Phillip Grimaldi
OpenStax Foundation
phillip.grimaldi@rice.edu

Harold Pashler
UCSD
hpashler@ucsd.edu

ABSTRACT

When engaging with a textbook, students are inclined to highlight key content. Although students believe that highlighting and subsequent review of the highlights will further their educational goals, the psychological literature provides no evidence of benefits. Nonetheless, a student's choice of text for highlighting may serve as a window into their mental state—their level of comprehension, grasp of the key ideas, reading goals, etc. We explore this hypothesis via an experiment in which 198 participants read sections from a college-level biology text, briefly reviewed the text, and then took a quiz on the material. During initial reading, participants were able to highlight words, phrases, and sentences, and these highlights were displayed along with the complete text during the subsequent review. Consistent with past research, the amount of highlighted material is unrelated to quiz performance. However, our main goal is to examine highlighting as a data source for inferring student understanding. We explored multiple representations of the highlighting patterns and tested Bayesian linear regression and neural network models, but we found little or no relationship between a student's highlights and quiz performance. Our long-term goal is to design digital textbooks that serve not only as conduits of information into the mind of the reader, but also allow us to draw inferences about the reader at a point where interventions may increase the effectiveness of the material.

Keywords

student modeling, bayesian regression, neural networks

1. INTRODUCTION

A premise of educational data mining is that the knowledge state of a student can be inferred by observation. However, knowledge state is opaque until students reach a level of understanding that they can be tested or they can solve problems. This delay makes interventions at an early stage of exposure quite challenging. Consider a student's first engagement with new material in a textbook. Reading times and gaze patterns may be useful for modeling student engagement and comprehension [3]. However, these implicit measures are quite difficult to collect. Fortunately, students often willingly provide explicit measures: students will voluntarily highlight sections of text and write notes in the margins. With the advent of electronic texts, the opportunity now exists to collect data from students during their early exposure to new material, and if knowledge state can be inferred, interventions can be performed early. In this article, we explore the hypothesis that these annotations—specifically highlights—can be used to predict comprehension, as assessed by a follow-up quiz.

Highlighting has been studied in the psychological literature from the perspective of whether highlighting is an effective study strategy. The current understanding is that the mere act of highlighting does not promote learning, nor does re-reading isolated sentences that were highlighted [1]. No relationship has been found between coarse statistics of highlighting (e.g. the total amount of text highlighted) and a student's performance/understanding [2].

In a few cases, highlighting has been shown to provide benefits. First, text which is pre-highlighted by an informed instructor can guide a student to focus on key content [4]. Second, restricting highlighting to encourage consideration of the material—e.g., permitting the student to highlight only one sentence per paragraph—can support understanding [5]. In contrast to this traditional literature that examines highlighting as a study tool, here we examine highlighting as a data source for inferring student understanding.

2. EXPERIMENT

We conducted an experiment in which participants read passages from a biology textbook. They later reviewed the passages, and then took a short quiz drawing on material from the passages. During initial reading, participants were allowed to highlight portions of the text (words, phrases, or sentences). These highlights were displayed along with the text during the review phase, and participants were instructed that highlighting could assist in the review.

2.1 Methodology

2.1.1 *Participants*

Participants aged 18 and above were recruited from Amazon Mechanical Turk. A total of 198 people completed the experiment and were paid $3.60. Data from six participants was discarded because these participants reported that they were unable to use the highlighting functionality in their web browser. The experiment took 25-30 minutes to complete. No screening was performed to determine an individual's background in biology. To incentivize attention to the task, participants were told that they would be entered into a raffle for a bonus prize of $15.00, with the number of entries equal to the number of correct reponses to the quiz questions.

2.1.2 *Materials*

Three passages were selected from the Openstax *Biology* textbook [7]. The passages were chosen with the expectation that they could be understood by a college-aged reader with no background in biology. The three passages concern the topic of sterilization, with one serving as an introduction, one discussing procedures, and the last summarizing commercial use. Twelve factual quiz questions were generated by selecting particular sentences from the passages and turning the factual statements in these sentences into fill-in-the-blank questions. These twelve questions were transformed into twelve additional multiple choice questions, each question comprised of the correct response and three lures as alternatives. Three questions are drawn from the first passage, four from the second passage, and five from the final passage.

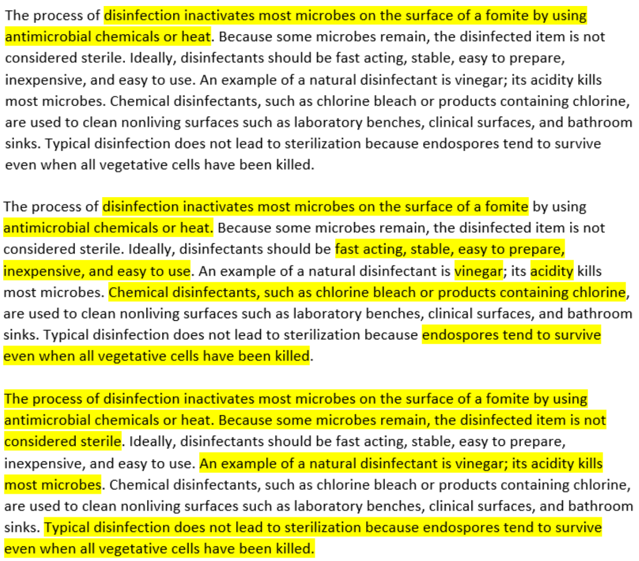
For each participant a *normalized quiz score* is computed as follows. For each of the twelve questions, a score of 1.0 is assigned if both the fill-in-the-blank and multiple-choice response are correct; a score of 0.66 is assigned if the fill-in-the-blank (FIB) response is correct; a score of 0.33 is assigned if the multiple-choice (MC) response is correct; and a score of 0 is assigned if neither is correct. The normalized quiz score is the sum of these scores divided by 12, yielding a value in the range [0,1]. A liberal criterion was used for judging FIB response correctness: A response is considered correct if the edit distance between the actual and correct responses is less than 25% of the length of the correct response. Table 1 shows the distribution of response correctness on MC and FIB versions of a question.

2.1.3 *Procedure*

The experiment is divided into three phases. During the *reading* phase, the three passages are presented on the screen sequentially, each on screen for five minutes. During the *review* phase, the three passages are again presented sequentially, along along with any highlights the participant made during the reading phase, each for one minute. Finally, during the *quiz* phase, the 12 FIB questions are shown, followed by the 12 MC questions, randomized within question type. During the first two phases, a timer at the top of the screen indicates time remaining for the current passage. After the timer has expired, the screen blanks and displays a message describing the next step of the experiment (either the next passage or the next phase of the experiment). Throughout the course of the experiment, a progress bar is displayed at the bottom of the screen that indicates the current proportion of the experiment completed.

Table 1: Distribution of response correctness on multiple choice (MC) and fill-in-the-blank (FIB) versions of a question

	MC Incorrect	MC Correct
FIB Incorrect	0.259	0.415
FIB Correct	0.038	0.288



The process of disinfection inactivates most microbes on the surface of a fomite by using antimicrobial chemicals or heat. Because some microbes remain, the disinfected item is not considered sterile. Ideally, disinfectants should be fast acting, stable, easy to prepare, inexpensive, and easy to use. An example of a natural disinfectant is vinegar; its acidity kills most microbes. Chemical disinfectants, such as chlorine bleach or products containing chlorine, are used to clean nonliving surfaces such as laboratory benches, clinical surfaces, and bathroom sinks. Typical disinfection does not lead to sterilization because endospores tend to survive even when all vegetative cells have been killed.

The process of disinfection inactivates most microbes on the surface of a fomite by using antimicrobial chemicals or heat. Because some microbes remain, the disinfected item is not considered sterile. Ideally, disinfectants should be fast acting, stable, easy to prepare, inexpensive, and easy to use. An example of a natural disinfectant is vinegar; its acidity kills most microbes. Chemical disinfectants, such as chlorine bleach or products containing chlorine, are used to clean nonliving surfaces such as laboratory benches, clinical surfaces, and bathroom sinks. Typical disinfection does not lead to sterilization because endospores tend to survive even when all vegetative cells have been killed.

The process of disinfection inactivates most microbes on the surface of a fomite by using antimicrobial chemicals or heat. Because some microbes remain, the disinfected item is not considered sterile. Ideally, disinfectants should be fast acting, stable, easy to prepare, inexpensive, and easy to use. An example of a natural disinfectant is vinegar; its acidity kills most microbes. Chemical disinfectants, such as chlorine bleach or products containing chlorine, are used to clean nonliving surfaces such as laboratory benches, clinical surfaces, and bathroom sinks. Typical disinfection does not lead to sterilization because endospores tend to survive even when all vegetative cells have been killed.

Figure 1: A paragraph of text as highlighted by three randomly selected participants.

In the reading phase, participants may highlight text by selecting one or more words using the mouse, which we will refer to as a highlighting *interaction*. If the selected text exactly corresponds to an existing highlight, the highlight is deleted. If the selected text captures any portion of an existing highlight but extends beyond it, the existing highlight is expanded to include the new selection. A single interaction may highlight more than one sentence at a time, but cannot cross paragraph boundaries. In the review phase, the previously selected highlights are displayed, but no additional highlights can be made.

3. RESULTS

Figure 1 presents an example of three participants' highlights of one paragraph of text. As these examples make clear, there is diversity in the manner in which individuals highlight. Highlights are used to note single words, phrases, and complete sentences.

In order to analyze the relationship between an individual's highlights and quiz performance, we need to first specify a representation of the highlights. In all analyses, we ignore the time course and sequence of actions that the participant took to create and/or delete highlights, and instead consider only the terminal highlighted state of each passage. The three passages contain 117 complete sentences delin-

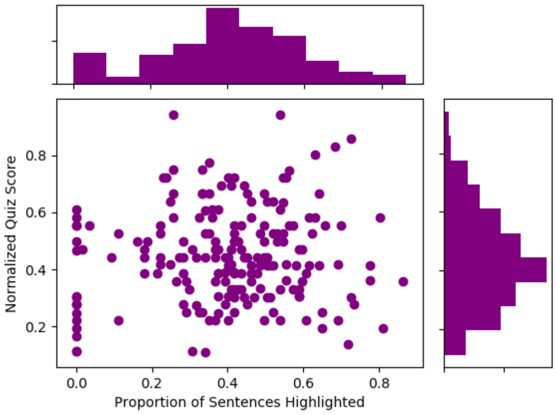


Figure 2: Scatter plot of proportion of sentences highlighted (using the binary encoding) versus normalized quiz score for each participant. The marginal distributions are shown above and to the right of the scatter plot.

eated by periods, exclamation marks, and question marks. The first analyses we perform are based on a *sentence-level* representation in which the pattern of highlights are coded as a 117-dimensional feature vector, either as a *binary* encoding in which each element i of the vector is set to 1 if any portion of sentence i is highlighted, or as a *graded* encoding in which element i is set to the proportion of words in the sentence that are highlighted.

Figure 2 shows the relationship between the proportion of sentences highlighted according to the binary encoding and the normalized quiz score. Each point is a single participant. As shown along the margin, the proportion of sentences highlighted is a unimodal distribution with a mean of 0.40. The normalized quiz score is also unimodal with a mean of 0.45. The scatter plot suggests no functional relationship—linear or otherwise—between the amount of highlighting and quiz performance; the correlation coefficient is 0.08.

Although the total number of highlights fails as a predictor of quiz score, the specific pattern of highlighting may prove more useful. To begin analyzing the relationship between highlighting patterns and performance, we performed a locally-linear embedding (LLE) with 11 neighbors [6] to reduce the dimensionality of the 117-dimensional binary sentence-level highlighting vector to a 2D space. Figure 3(a) plots the embedded points, colored to indicate the corresponding quiz score. The embedding has interesting structure, but no simple relationship to quiz performance. To understand what the LLE has captured, the points are recolored by proportion of sentences highlighted in Figure 3(b). This figure reveals that the abscissa captures the proportion, and the ordinate captures some of the diversity in the representation for a particular proportion. Referring back to Figure 3(a), there is no discernible relationship between the variation along the ordinate and performance, even when there is diversity in the embedding (i.e., the mid-range along the abscissa).

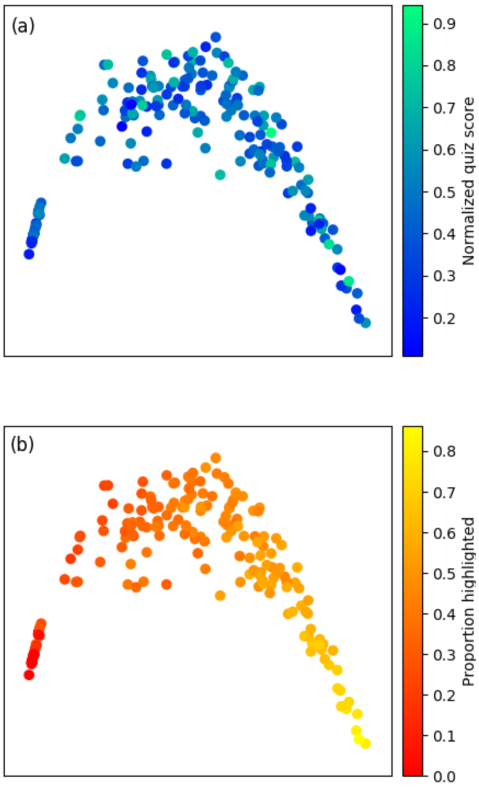


Figure 3: 2D LLE embedding of the binary sentence-level highlights with each point corresponding to one participant's data, and the coloring of points indicating (a) normalized quiz score and (b) the proportion of sentences highlighted.

We explored other parameterizations of LLE and other dimensionality reduction methods (e.g., k means clustering) but found no discernible relationship between performance and the reduced representations.

3.1 Modeling results

We constructed a series of models that map the highlighted-sentence representation—either the binary or graded encoding—to either total quiz score or correctness on specific problems. In all model testing, we perform nested cross validation to optimize model hyperparameters and evaluate model generalization to new participants. Our nested procedure consists of an outer 10-fold cross validation loop to partition the entire data set by participants into training and test sets, and an inner 3-fold cross validation loop further splitting the training set to select hyperparameters. The best set of hyperparameters chosen from the inner loop are selected and the entire training set is then used to build a model which predicts test set performance. This process is repeated over the outer loop to obtain an average normalized model loss.

The normalized model loss is defined as:

$$L = \frac{\mathrm{E}_i[\mathrm{E}_j[(s_{ij} - \hat{s}_{ij})^2]]}{\mathrm{E}_i[\mathrm{E}_j[(s_{ij} - \bar{s}_i)^2]]},$$

where s_{ij} is the test score of participant j in outer fold i, $\hat{s}_{ij}$ is the corresponding model prediction, $\bar{s}_i$ is the mean score of the participants in the training set for fold i, and E_i and E_j are the expectations taken over folds and participants, respectively. This normalized loss is 1.0 if the model does no better than predicting the mean score of the training participants, and drops to 0.0 if predictions are perfect. The proportion of variance explained by the model is $1 - L$.

3.1.1 Linear models

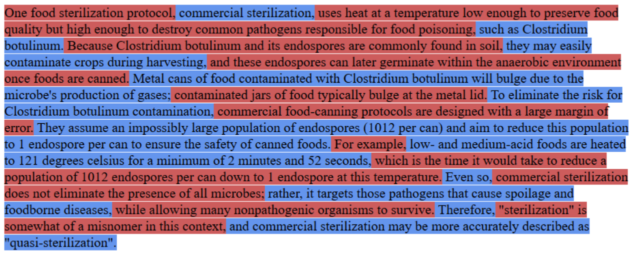
The first set of models we examine are based on Bayesian linear regression. The general form of these models is $y = Wx+b$, where y is the predicted quiz score, x is the highlight vector, and W and b are free parameters. This variant of linear regression is well suited for domains in which the number of input features is high relative to the volume of data. The model coefficients are regularized via a prior which achieves a ridge penalty (a penalty for large weights). The model is specified with a prior distribution on the precision of the observation noise, Gamma($shape = 10^{-\kappa_1}, rate = 10^{-\kappa_2}$) and prior on the precision of the coefficients, Gamma($shape = 10^{-\kappa_3}, rate = 10^{-\kappa_4}$), where $\kappa_* \in \{3, 4, 5, 6, 7, 8, 9\}$ was chosen by the inner cross validation loop.

Consistent with the scatter plot in Figure 2, when predicting on overall quiz performance, we found that a regression on the total number of sentence-level highlights obtains a mean normalized loss of 1.01 (SEM 0.0029). Similarly, a regression on the total number of words highlighted obtains a mean normalized loss of 1.01 (SEM 0.0028). Table 2 summarizes these and subsequent results.

Although the summary statistics fail to predict performance, one might hope to see a relationship between the specific *pattern* of highlights and performance. Unfortunately, regressions on the binary and graded sentence-level representations of highlights obtain mean normalized losses of 0.99 (SEM 0.0028) and 1.03 (SEM 0.0032), respectively.

We hypothesized that the sentence-level representations of highlights may be too coarse to capture important differences in the highlighting patterns. We therefore parsed the text based on a *sentence-fragment* representation in which the passages are segmented by periods, exclamation marks, question marks, colons, semicolons, as well as phrase-separating commas. The inclusion or exclusion of commas as segment boundaries was subjective; our strategy was to exclude commas that were used to delineate lists of items. Figure 4 gives an example of the sentence-fragment partition scheme. This partition scheme yields 235 sentence fragments across the three passages. We examined both binary and graded representations of the fragment-level highlights. The regression on the binary and graded fragment-level representations yields mean normalized losses of 0.93 (SEM 0.0024) and 1.03 (SEM 0.0032), respectively.

Parsing the passages at an even finer granularity, we constructed a representation of the individual *words* highlighted. The three passages have a total of 2291 word tokens. A regression on this raw representation of highlights yields a mean normalized loss of 0.93 (SEM 0.0024).



One food sterilization protocol, commercial sterilization, uses heat at a temperature low enough to preserve food quality but high enough to destroy common pathogens responsible for food poisoning, such as Clostridium botulinum. Because Clostridium botulinum and its endospores are commonly found in soil, they may easily contaminate crops during harvesting, and these endospores can later germinate within the anaerobic environment once foods are canned. Metal cans of food contaminated with Clostridium botulinum will bulge due to the microbe's production of gases; contaminated jars of food typically bulge at the metal lid. To eliminate the risk for Clostridium botulinum contamination, commercial food-canning protocols are designed with a large margin of error. They assume an impossibly large population of endospores (1012 per can) and aim to reduce this population to 1 endospore per can to ensure the safety of canned foods. For example, low- and medium-acid foods are heated to 121 degrees celsius for a minimum of 2 minutes and 52 seconds, which is the time it would take to reduce a population of 1012 endospores per can down to 1 endospore at this temperature. Even so, commercial sterilization does not eliminate the presence of all microbes; rather, it targets those pathogens that cause spoilage and foodborne diseases, while allowing many nonpathogenic organisms to survive. Therefore, "sterilization" is somewhat of a misnomer in this context, and commercial sterilization may be more accurately described as "quasi-sterilization".

Figure 4: Example of sentence-fragment representation where the alternating colors signify the different fragments.

Although we obtain a modest (7%) reduction in variance with the binary fragment-level highlights and the individual-word highlights, it is likely that these seemingly promising results are meaningless because the model degrees of freedom (235 and 2291, respectively) are larger than the number of subjects in our data set (198).

Because our models have sufficient degrees of freedom that they are not well constrained by the data, we turn to simple models that leverage domain knowledge, specifically, our knowledge of which sentence in the text contains the critical information for a given quiz question. We tested the correlation between highlighting a critical sentence and improved performance on the corresponding quiz question. Analyzing the fill-in-the-blank (FIB) and multiple choice (MC) questions separately, a two-tailed matched-sample t-test indicates that MC quiz scores are significantly higher for those who highlighted the critical sentence than for those who did not (0.74 versus 0.63, $t(11) = 4.05$, $p = 0.002$, $d = 0.73$). A marginal effect in the expected direction was also found for FIB by those who highlighted the critical sentence versus those who did not (0.34 versus 0.29, $t(11) = 2.034$, $p = 0.067$, $d = 0.23$).

We then built linear regression models to determine how the conditional analysis of the previous paragraph translates to predictive model performance. Models were built predicting specific quiz question accuracy from the corresponding critical sentence, separately for MC and FIB and for each of the 12 questions. Models were evaluated using 10-fold cross validation. Averaging across the 12 questions, the normalized loss is 0.99 for MC, ranging over questions from 0.94 to 1.02, and the normalized loss is 1.00 for FIB, ranging over questions from 0.95 to 1.03.

Although pre-selecting the critical passage elements does not appear to boost prediction, we are optimistic, based on the conditional probability analysis, that with additional data, our models will begin to reveal dependencies.

In conclusion, none of the linear models are particularly promising. Although two of the models do seem to predict some variance in the test scores—models utilizing the binary fragment-level and the word-level representations—one has to be cautious in reaching a positive conclusion given the large number of models we constructed.

3.1.2 Nonlinear models

We also evaluated nonlinear regression models, specifically neural networks. The neural networks had one or two hidden layers with tanh activation functions and an output layer with a single sigmoid unit to represent the normalized test score prediction. The nets were trained by the Adam optimizer to minimize the mean square error between the normalized quiz score and the prediction, with an initial learning rate of 0.001 and batch size equal to 20% of the size of the training set. All weights were initially drawn using Xavier initialization. A validation set was created from 10% of the supplied training data, which was used to stop training after the normalized error on the validation set plateaued after 50 epochs. Model hyperparameters (see table below) were chosen by a grid search in the inner cross validation loop. The regularizers include an L2 weight penalty on the input-to-hidden weights and dropout on the nodes in the hidden layers.

Grid Search	
Hyper Parameter	Values
Dropout rate	0, 0.5
Hidden layer 1 size	5, 10, 15, 20
Hidden layer 2 size	0, 5, 10, 15
L2 regulariz. relative learn rate	0, 0.25, 0.5, 0.75, 1

For each highlight representation (sentence-level, sentence-fragment, individual words), we found the best hyperparameters over the grid search and evaluated the models using 10-fold cross validation. We present the results of each of these networks in Table 3. Unfortunately, none of these models outperformed the baseline.

We hypothesized that there might be information to leverage by predicting performance on individual questions rather than their sum (the total quiz score). We therefore built neural net models with outputs that represent the individual questions, with two output units for each of the 12 questions. The target tuple (0,0) represents neither fill-in-the-blank (FIB) nor multiple-choice (MC) response correct; (0,1) represents FIB incorrect but MC correct; (1,0) represents FIB correct but MC incorrect; and (1,1) represents both FIB and MC correct. The logic of this coding scheme is that the first bit indicates strong knowledge of the answer and the second bit indicates at least weak knowledge.

The training and evaluation process was the same as the neural networks that predict on overall quiz score, with the normalized loss an expectation over the 24 outputs. We evaluated networks for each of the highlight representations, and Table 3 lists the results. Unfortunately, no predictions were better than baseline.

4. DISCUSSION

If you pick up any textbook in a used bookstore, you'll be surprised if it isn't marked up with student annotations and highlights. Students seem compelled to highlight because they believe it supports learning. Our goal was to leverage this compulsion to better understand what students are learning from their textbooks. We hypothesized that a learner's choice of material for highlighting could differentiate among individuals and predict comprehension. We constructed a wide range of models that use the specific pattern of highlights to predict subsequent quiz performance and specific quiz answers, yet we failed to obtain strong support for our hypothesis.

The most generous interpretation of our modeling effort is that when highlights are represented at a fine-level of granularity—sentence fragments or individual words—linear models can predict about 6% of the variability in quiz score. It's difficult to explain why the linear models (Table 2) outperform the nonlinear models with the same input representation, but perhaps we are not successfully controlling for overfitting of the more complex models. The variance in model predictions across cross-validation folds is an indication that the models are perhaps still too flexible and would benefit by stronger regularization.

The present experiment had several sources of uncontrolled variability that, in retrospect, should have been taken into account.

- We neglected to ask participants about their familiarity with biology and we did not exclude participants based on their knowledge. Prior knowledge could be a significant uncontrolled factor. In subsequent experiments, it would be sensible to screen participants based on whether they have had a biology class in the past three years.
- The randomized order of quiz questions influences the interval of time for which knowledge must be retained. For example, if the first quiz question is on the third passage of text, then the lag between reviewing that passage and the quiz question is just a matter of seconds. It would be more sensible to present the quiz questions in order by section and to randomize the order within a passage.
- In the present experiment, participants had little idea of what the quiz would entail until they completed the initial reading and review stages of all three passages. We suspect that participants may highlight in a more informed manner if they can better anticipate what is to come in the experiment. Thus, we might have included in the instructions a sample paragraph and several typical exam questions.
- We encouraged participants to highlight, but we did not ask participants whether they ordinarily highlight text as they read. There seems to be individual differences in the proclivity to highlight, and it would be useful to perform analyses of the highlights for the subpopulations who either do or do not ordinarily highlight.

A natural thought for improving predictive models is to encode information about the content of the text and semantic relationships among the individual sentences and phrases. We argue that such encodings will *not* improve our models for the specific experiment we have performed. If our goal was to devise a general passage-independent representation of text, then incorporating such encodings would be critical, but because we have three specific passages and our highlight representation allows for the reconstruction of which

Table 2: Summary of linear regression results

Input Features	Target Output	Mean Normalized Loss	Standard Error of Mean
Total number of sentence-level highlights	Normalized Quiz Score	1.01	0.0029
Total number of words highlighted	Normalized Quiz Score	1.01	0.0028
Binary sentence-level highlights	Normalized Quiz Score	0.99	0.0028
Graded sentence-level highlights	Normalized Quiz Score	1.03	0.0032
Binary sentence-fragment highlights	Normalized Quiz Score	0.93	0.0024
Graded sentence-fragment highlights	Normalized Quiz Score	1.03	0.0032
Word-level highlights	Normalized Quiz Score	0.93	0.0024
Critical-sentence highlight	Corresponding FIB Question Score	1.00	N/A
Critical-sentence highlight	Corresponding MC Question Score	0.99	N/A

Table 3: Summary of neural network results

Input Features	Target Output	Mean Normalized Loss	Standard Error of Mean
Binary sentence-level highlights	Normalized Quiz Score	1.01	0.0030
Graded sentence-level highlights	Normalized Quiz Score	1.00	0.0022
Binary sentence-fragment highlights	Normalized Quiz Score	0.99	0.0026
Graded sentence-fragment highlights	Normalized Quiz Score	1.20	0.0032
Word-level highlights	Normalized Quiz Score	1.03	0.0021
Binary sentence-level highlights	Individual Question Scores	1.00	0.0049
Graded sentence-level highlights	Individual Question Scores	1.00	0.0052
Binary sentence-fragment highlights	Individual Question Scores	1.00	0.0054
Graded sentence-fragment highlights	Individual Question Scores	1.00	0.0053
Word-level highlights	Individual Question Scores	1.00	0.0050

specific sentences, phrases, or words were highlighted, we argue that this representation is sufficient for prediction. For example, if the participant were to highlight all phrases related to thermal death time, we do not need an explicit representation of this concept because the pattern of sentences highlighted contains this information implicitly.

We have ideas for extending the present work with the hope that highlighting might serve as a valuable data source for inferring student knowledge. We mention several key ideas here.

- We explored a variety of highlighting representations in order to capture critical differences among highlighting patterns. However, we are not convinced that all critical differences are captured. Consider the following sentence from one of the passages in the experiment: *Unlike disinfectants, antiseptics are antimicrobial chemicals safe for use on living skin or tissues.* Highlights of this sentence in our data set include:

  - *antiseptics*
  - *antiseptics are antimicrobial chemicals*
  - *antiseptics are antimicrobial chemicals safe for use on living skin or tissues.*

  All three of these highlights are treated the same by the sentence and fragment representations with the binary encoding, but one might imagine that they provide different windows into the student's intentions. The individual word representation does distinguish these patterns, though at the cost of a much larger input and parameter space. The sentence-level and sentence-fragment graded encodings seem to be a sensible intermediate, but we suspect there are other intermediate encodings that would be fruitful to explore.

- One potentially useful source of information would be the detailed time course of reading, i.e., fixation patterns as a function of time, or at least obtaining information on the rate at which sentences are read and when backtracking occurs. In our current experiment, timing information is recorded only when a sentence is highlighted; these data are too sparse to provide a useful representation that can be compared across individuals.

  In order to record better timing information, we have considered conducting the experiment using a small screen e-reader (or a small window on a computer monitor) which necessitates scrolling from one paragraph to the next.

5. ACKNOWLEDGMENTS

This research was supported by the National Science Foundation award EHR-1631428.

6. REFERENCES

[1] John Dunlosky, Katherine A Rawson, Elizabeth J Marsh, Mitchell J Nathan, and Daniel T Willingham. 2013. Improving students' learning with effective learning techniques: Promising directions from cognitive and educational psychology. *Psychological Science in the Public Interest* 14, 1 (2013), 4–58.

[2] Robert L Fowler and Anne S Barker. 1974. Effectiveness of highlighting for retention of text material. *Journal of Applied Psychology* 59, 3 (1974), 358.

[3] Caitlin Mills, Art Graesser, Evan F Risko, and Sidney K D'Mello. 2017. Cognitive coupling during reading. *Journal of Experimental Psychology: General* 146, 6 (2017), 872.

[4] Sherrie L Nist and Mark C Hogrebe. 1987. The role of underlining and annotating in remembering textual information. *Literacy Research and Instruction* 27, 1 (1987), 12–25.

[5] John P Rickards and Gerald J August. 1975. Generative underlining strategies in prose recall. *Journal of Educational Psychology* 67, 6 (1975), 860.

[6] Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science* 290, 5500 (2000), 2323–2326.

[7] Connie Rye, Robert Wise, Vladamir Jurukovski, Jung Desaix, and Yael Avissar. 2016. *Biology.* OpenStax.

An Empirical Research on Identifiability and Q-matrix Design for DINA model

Peng Xu
Polytechnique Montreal
peng.xu@polymtl.ca

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

ABSTRACT

In most contexts of student skills assessment, whether the test material is administered by the teacher or within a learning environment, there is a strong incentive to minimize the number of questions or exercises administered in order to get an accurate assessment. This minimization objective can be framed as a Q-matrix design problem: given a set of skills to assess and a fixed number of question items, determine the optimal set of items, out of a potentially large pool, that will yield the most accurate assessment. In recent years, the Q-matrix identifiability under DINA/DINO models has been proposed as a guiding principle for that purpose. We empirically investigate the extent to which identifiability can serve that purpose. Identifiability of Q-matrices is studied throughout a range of conditions in an effort to measure and understand its relation to student skills assessment. The investigation relies on simulation studies of skills assessment with synthetic data. Results show that identifiability is an important factor that determines the capacity of a Q-matrix to lead to accurate skills assessment with the least number of questions.

1. INTRODUCTION

Consider a set of items intended to assess a student's mastery over a set of skills, or knowledge components (KC). These items, along with the set of skills, can be designed to test a single skill at once. Or, they can be designed to involve two or more skills. A test composed of a fixed number of items can either be composed of a mixture of single and multiple skills items, or composed of one type of items only. Skills can themselves be defined so as to facilitate the creation of task/problem items that involve single skill per item, or multiple skills per items. By which principles should a teacher choose among these different options?

This paper addresses this question, with the general objective of designing a test that will bring the most accurate assessment of a student's skill mastery state with the least number of questions items.

The investigation is framed within the DINA model, which was a widely researched model and originally proposed in the research of a rule space method for obtaining diagnostic scores (Tatsuoka, 1983). In this model, question items can involve one or more skills, and all skills are required in order to succeed the question, while a success can still occur through a guessing factor, and failure can also occur through a slip factor.

2. Q-MATRIX, DINA MODEL AND IDENTIFIABILITY

The mapping of items to skills is referred to as a Q-matrix, where items are mapped to latent skills whose mastery is deemed necessary in order for the student to succeed at the items. An item can represent a question, an exercise, or any task that can have a positive or negative outcome. In the DINA model, the conjunctive version of the Q-matrix is adopted: all skills are considered necessary for success.

In the last decade, a number of papers have been devoted to deriving a Q-matrix from student test results data (Barnes, 2010; Liu, Xu, & Ying, 2012; Desmarais, Xu, & Beheshti, 2015; P. Xu & Desmarais, 2016). Another line of research on Q-matrices has been devoted to refine or to validate an expert-given Q-matrix (de la Torre & Chiu, 2015; Chiu, 2013; Desmarais & Naceur, 2013). While the problems of deriving or refining a Q-matrix from data are related to Q-matrix design, they do not provide insight into how best to design them.

In parallel to these investigations, some researchers have looked at the question of the identifiability. The general idea behind identifiability is that two or more configurations of model parameters can be considered as equivalent. Sets of parameters will be considered equivalent if, for example, their likelihood is equal given a data sample. Or, conversely, if the parameters are part of a generative model, two sets of equivalent parameters would generate data having the same characteristics of interest, in particular equal joint probability distributions (see Doroudi & Brunskill, 2017, for more details).

The issue of identifiability for student skills assessment was first researched in multiple diagnosis model comparison (Yan, Almond, & Mislevy, 2004), Bayesian Knowledge Tracing (Beck & Chang, 2007) and later discussed by more researchers (van De Sande, 2013; Doroudi & Brunskill, 2017). A mathematically rigorous treatment Q-matrix identifiability under the DINA/DINO setting was presented under zero slip and guess parameters (Chiu, Douglas, & Li, 2009), and under known slip and guess (Liu, Xu, & Ying, 2013), and finally under unknown slip and guess parameters (Chen, Liu, Xu, & Ying, 2015). An overall discussion can also be found (G. Xu & Zhang, 2015; Qin et al., 2015). These studies provide theoretical basis to derive Q-matrices from data, but not to the design of Q-matrices itself. In this paper, we consider the identifiability of the Q-matrix with regards to the DINA model.

Identifiability is a general concept for statistical models. Its formal definition is:

Definition (1) (Casella & Berger, 2002) A parameter θ for a family of distribution $f(x|\theta : \theta \in \Theta)$ is *identifiable* if distinct values of θ correspond to distinct pdfs or pmfs. That is, if $\theta \neq \theta'$, then $f(x|\theta)$ is not the same function of x as $f(x|\theta')$.

The DINA model has parameters $\theta = \{Q, p, s, g\}$, where Q is the Q-matrix. p is the categorical distribution parameter for all student profile categories. That is, it indicates the probability that a student belongs to each profile category. For example, in a 3-skill case, there are $2^3 = 8$ categories for students to belong to, and the 8-component probability vector of students belongs to each of these categories is the model parameter p. Finally, s and g are both vectors denoting the slip and guess of each item.

The identifiability of all parameters in DINA model have been thoroughly investigated and several theorems are given (G. Xu & Zhang, 2015). But for the Q-matrix design problem that is the focus of this paper, we solely need to ensure that the model parameter p is identifiable, meaning that we can distinguish different profile categories. Fortunately, for the case when s and g are known, the requirement is easily satisfied, since it only requires the Q-matrix to be *complete*.

Definition (2) (Chen et al., 2015) The matrix Q is *complete* if $\{e_i : i = 1, ..., K\} \subset R_Q$, where K is the number of skills (columns of Q), R_Q is the set of row vectors of Q, and e_i is a row vector such that the i-th element is one and the rest are zero (i.e. a binary unit vector, also known as a "one-hot vector"). Stated differently, the rows of the identity matrix, $I_{K\times K}$, must be in Q for this matrix to be complete.

And the heart of the current investigation is based on the following proposition:

Proposition (Chen et al., 2015) Under the DINA and DINO models, with Q, s and g being known, the population proportional parameter p is *identifiable* if and only if Q is *complete*.

We show an example of Q-matrix that is not complete below for better illustration.

$$\begin{array}{c|ccc} & k_1 & k_2 & k_3 \\ \hline q_1 & 1 & 0 & 0 \\ q_2 & 0 & 1 & 1 \\ q_3 & 1 & 0 & 1 \end{array}$$

This Q-matrix does not contain $e_2 : [0, 1, 0]$ or $e_3 : [0, 0, 1]$, and is therefore not complete, even though its items (rows) cover all skills (columns). Using this Q-matrix under DINA model setting entails that the model parameters are not identifiable according to the proposition above, and would in turn compromise student profile diagnosis. In fact, students who only master skill 2 and students who only master skill 3 are indistinguishable under this Q-matrix.

But while the use of a non identifiable Q-matrix should be avoided according to the proposition, the question remains: among all the complete Q-matrix, which ones are most efficient for student profile diagnosis?

In the next section, we investigate empirically the Q-matrix design options in light of the *completeness* requirement, using synthetic student performance data with the DINA model. Synthetic data is essential for this investigation because we need to know the underlying ground truth. We return to the issue of using real data in the conclusion.

3. EXPERIMENT

The Q-matrix design problem is essentially an optimization problem. Basically, we have a pool of Q-matrices, and each of them is formed by a selection with replacement from a pool of q-vectors. Each Q-matrix will yield some capacity to diagnose students, as measured by a loss function. We aim to choose a Q-matrix that minimizes the loss function.

Our experiments follow a Bayesian framework to diagnose students under DINA Q-matrices. First, we use one-hot encoding to denote all profile categories. Set M to be the number of profile categories. Then, in the 3-skill case, the $M = 8$ profile categories pc_i are:

$$\begin{array}{c|ccc} & k_1 & k_2 & k_3 \\ \hline pc_1 & 0 & 0 & 0 \\ pc_2 & 1 & 0 & 0 \\ pc_3 & 0 & 1 & 0 \\ pc_4 & 0 & 0 & 1 \\ pc_5 & 1 & 1 & 0 \\ pc_6 & 1 & 0 & 1 \\ pc_7 & 0 & 1 & 1 \\ pc_8 & 1 & 1 & 1 \end{array}$$

Therefore, a student belonging to profile pc_1 is encoded as a binary unit vector $\alpha_1 = (1, 0, 0, 0, 0, 0, 0, 0)$, and so on for pc_2 encoded as $\alpha_2 = (0, 1, 0, 0, 0, 0, 0, 0)$, ..., and pc_8 encoded as $\alpha_8 = (0, 0, 0, 0, 0, 0, 0, 1)$. The DINA model parameter p is represented as a probability vector $p = (p_1, p_2, ..., p_8) = (P(\alpha_1), P(\alpha_2), ..., P(\alpha_8))$. Then, we set the prior of each student profile to be:

$$\alpha_0 = (1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8)$$

With the conditional independence assumed (i.e, conditioned on a given profile category, the probability to answer each question correct is independent), the likelihood is given by (De La Torre, 2009; Chen et al., 2015):

$$\begin{aligned} L(p, Q, s, g|X) &= P(X|p, Q, s, g) \\ &= \prod_{i=1}^{I} \sum_{\alpha} p_\alpha P(X_i|\alpha, Q, s, g) \\ &= \prod_{i=1}^{I} \sum_{\alpha} p_\alpha \prod_{j=1}^{J} P_j(\alpha)^{X_{ij}} [1 - P_j(\alpha)]^{1-X_{ij}} \end{aligned} \tag{1}$$

in which X is the response matrix and X_i is the i-th row, I is the number of records (students), J is the number of questions. $P_j(\alpha)$ is the probability of student profile α to answer correctly of question j, notice α in 3-skill case has only 8 possible values, for any of them $\alpha_m, m = 1, ..., 8$, the probability is given by DINA model

$$P_j(\alpha_m) = P(X_{ij} = 1|\alpha_m) = g_j^{1-\eta_{mj}} (1 - s_j)^{\eta_{mj}}$$

in which η_{mj} is the latent response of profile α_m to question j, that is, the response when slip and guess is 0. It can be calculated by

$$\eta_{mj} = \prod_{k=1}^{K} \alpha_{mk}^{q_{jk}}$$

where K is the number of skills and q_{jk} is the (j,k)-th element of Q-matrix Q.

Given the prior and likelihood, the posterior $\hat{\alpha}$ for each student can be calculated. It has the form:

$$\hat{\alpha} = (\hat{p}_1, \hat{p}_2, \hat{p}_3, \hat{p}_4, \hat{p}_5, \hat{p}_6, \hat{p}_7, \hat{p}_8)$$

and we then calculate the loss between this posterior and the true profile α_{true}, which is one of the one-hot encoding vector.

For any Q-matrix configuration, the loss function is defined by

$$loss(Q) = \sum_{i \in \text{students}} \|\hat{\alpha}_i - \alpha_{\text{true}}\|^2$$

To implement the experiment, for each Q-matrix configuration, we generate a response matrix based on the DINA model given fixed slip and guess parameters, using function 'DINAsim' from the R package *DINA* (Culpepper, 2015). Then, we calculate the posterior estimation for all students and evaluate the total loss. The reported result is an average loss of 100 runs.

In our experiments, we consider the 3-skills and 4-skills cases. For the 3-skills case, experiments are conducted with $N = 200$ students, of which 25 students fall into each of 8 categories. For the 4-skills case, we use $N = 400$ students, of which 25 students fall into each of 16 categories.

3.1 Experiment 1: Comparison of three strategies

In the first experiment, we compare three different Q-matrix design strategies. They are all based on repetition of a specific pool of q-vectors.

- Strategy 1 (Q-matrix 1): Using the identifiability condition (definition (1)) by using only combinations of the vectors $\{e_i : i = 1, ..., K\}$ (binary unit vectors, or one-hot encodings).
- Strategy 2 (Q-matrix 2): Using the vectors $\{e_i : i = 1, ..., K\}$ plus an all-one vector $(1, 1, 1)$ (in 3-skill case) or $(1, 1, 1, 1)$ (in 4-skill case). This is inspired by orthogonal array design, which is a commonly seen design of experiments (Montgomery, 2017).
- Strategy 3 (Q-matrix 3): Repeatedly using all q-vectors.

For the 3-skills case, all these three Q-matrices are shown in Figure 1. The general pattern is to recycle the rows above the lines denoted by $...[..., ..., ...]$.

The 4-skills case is similar, which is omitted here. Results of these two cases are shown in Figure 2a and Figure 2b.

Q-matrix 1
(binary unit vectors)

$$\begin{array}{c|ccc|} & k_1 & k_2 & k_3 \\ q_1 & 1 & 0 & 0 \\ q_2 & 0 & 1 & 0 \\ q_3 & 0 & 0 & 1 \\ \cdots & \cdots & \cdots & \cdots \\ q_{19} & 1 & 0 & 0 \\ q_{20} & 0 & 1 & 0 \\ q_{21} & 0 & 0 & 1 \end{array}$$

Q-matrix 2
(binary unit + all-1s vectors)

$$\begin{array}{c|ccc|} & k_1 & k_2 & k_3 \\ q_1 & 1 & 0 & 0 \\ q_2 & 0 & 1 & 0 \\ q_3 & 0 & 0 & 1 \\ q_4 & 1 & 1 & 1 \\ \cdots & \cdots & \cdots & \cdots \\ q_{17} & 1 & 0 & 0 \\ q_{18} & 0 & 1 & 0 \\ q_{19} & 0 & 0 & 1 \\ q_{20} & 1 & 1 & 1 \\ q_{21} & 1 & 0 & 0 \end{array}$$

Q-matrix 3
(all combinations)

$$\begin{array}{c|ccc|} & k_1 & k_2 & k_3 \\ q_1 & 1 & 0 & 0 \\ q_2 & 0 & 1 & 0 \\ q_3 & 0 & 0 & 1 \\ q_4 & 1 & 1 & 0 \\ q_5 & 1 & 0 & 1 \\ q_6 & 0 & 1 & 1 \\ q_7 & 1 & 1 & 1 \\ \cdots & \cdots & \cdots & \cdots \\ q_{15} & 1 & 0 & 0 \\ q_{16} & 0 & 1 & 0 \\ q_{17} & 0 & 0 & 1 \\ q_{18} & 1 & 1 & 0 \\ q_{19} & 1 & 0 & 1 \\ q_{20} & 0 & 1 & 1 \\ q_{21} & 1 & 1 & 1 \end{array}$$

Figure 1: Q-matrix design strategies

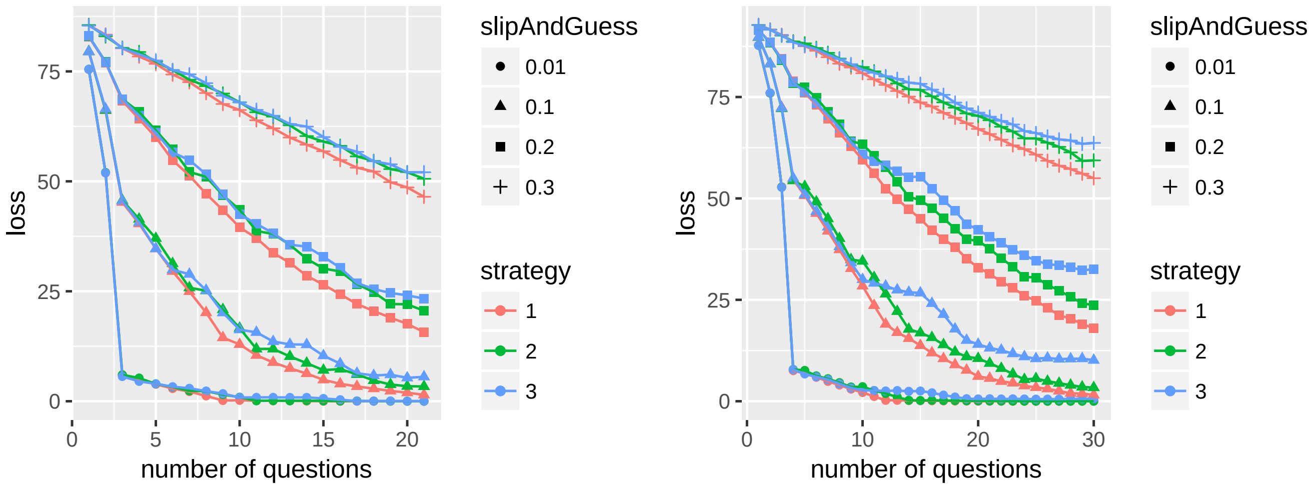
3.2 Experiment 2: Find best configuration

The second experiment takes the brute force approach. We directly examine all possible Q-matrix configurations. First, for a given pool of q-vectors to choose from and an integer indicating the number of questions, we need to know the number of possible configurations of Q-matrices we have. This is equivalent to a classical combinatorial problem, that is, to allocate marbles (q-vectors) to bins (questions). It can be easily computed by combinatorial coefficients and interpreted by using stars and bars methods. For example, in 3-skills case, we have 7 q-vectors, and if we have 4 questions to allocate them, then we have $\binom{4+7-1}{7-1} = 210$ possible configurations. This number grows up sharply as a number of questions increases or number of patterns increases. As a comparison, in the 4-skills case, if we have 5 questions to allocate them, then we have $\binom{5+15-1}{15-1} = 11628$ possible configurations.

For each configuration, we calculate the MAP estimation for all categories of each student, and compare with the one-hot encoding for their true categories. The total loss is reported as the performance index.

Figure 3 shows the results of 6 combinations of different numbers of skills and questions:

- 3-skills case, 4 questions: Figure 3a, Figure 3b
- 3-skills case, 8 questions: Figure 3c, Figure 3d
- 4-skills case, 5 questions: Figure 3e, Figure 3f



(a) 3-skills case

(b) 4-skills case

Figure 2: Experiment 1: Three Strategy Comparison on 3- and 4-skills cases

4. DISCUSSION

From the result of experiment 1 we can see that strategy 1 always works better than the other two strategies, meaning that simply repeating the vectors $\{e_i : i = 1, ..., K\}$ in Q-matrix design, without using any combination of skills, yields better student diagnosis performance.

From the result of experiment 2, when slip and guess parameters are as low as 0.01, we can see obvious graded patterns among different configurations. This can be explained by the the distinguishability of a Q-matrix. For example, in Figure 3a, we can see there are 7 layers. In fact, the first layer consisted of Q-matrix that can only cluster students into 2 categories. One example of such a Q-matrix is

$$\begin{array}{c} \\ q_1 \\ q_1 \\ q_1 \\ q_1 \end{array} \begin{array}{c} \begin{array}{ccc} k_1 & k_2 & k_3 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \end{array}$$

This Q-matrix can only discriminate between a student that mastered skill 1 or not. We know that there are in fact 8 categories of students, the 7 layers in Figure 3a from top to bottom correspond to the Q-matrix that can separate students into 2 to 8 categories. We can see that complete Q-matrices always fall in the bottom layer, which concurs with the proposition of Section 2. The 4-skills case is similar in Figure 3e.

When slip and guess parameter increase, the points become more divergent, as can be seen by comparison between figures 3a and 3b. In order to see some greater details, we distinguish three types of Q-matrices.

- Type I: Complete and confined, meaning it is only consisted of vectors $\{e_i : i = 1, ..., K\}$.
- Type II: Complete but not confined, meaning it not only contains all vectors $\{e_i : i = 1, ..., K\}$, but also contains at least one other q-vector.
- Type III: Incomplete Q-matrix.

Type I and Type II Q-matrices performs the same when slip and guess are low (figures 3a, 3e), but when they get higher, Type I Q-matrices show a better performance (figures 3b, 3f).

However, when more questions are involved in a high slip and guess condition, the performance becomes more unstable. Therefore, we again consider more subtypes. In 3-skills case for 8 questions, we consider three subtypes below.

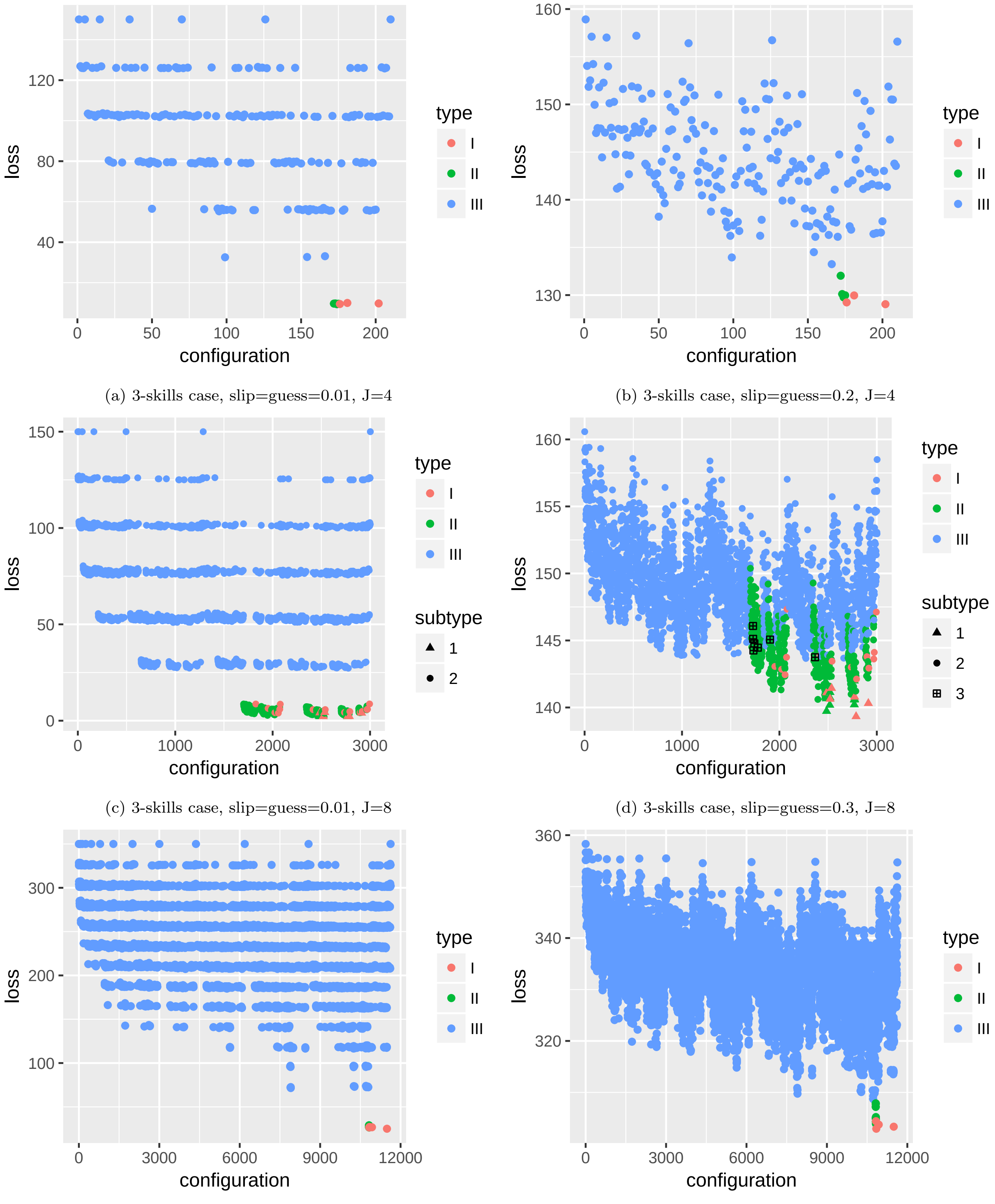
- Subtype 1: Q-matrix contains each component of $\{e_i : i = 1, ..., K\}$ at least twice.
- Subtype 2: Other situations (e.g A complete Q-matrix but all the other vectors are just repeated e_1).
- Subtype 3: Q-matrix contains all q-vectors.

From Figure 3d we can see that the subtype 1 (denoted by triangle) shows better performance than subtype 2, meaning that repeating the whole set of $\{e_i : i = 1, ..., K\}$ is a better strategy just like the strategy 1 we used in experiment 1. Subtype 3 corresponds to the strategy 3 in experiment 1, it has only 7 possible configurations in 8-question setting and we can see that they do not perform well.

Therefore, we argue that the best Q-matrix design is to use only the vectors $\{e_i : i = 1, ..., K\}$ since it offers quicker convergence speed (as shown in experiment 1) and better robustness against slip and guess (as shown both in experiments 1 and 2).

5. CONCLUSION

This work is still in an early stage and has limitations, in particular because it is conducted with synthetic data. But



(a) 3-skills case, slip=guess=0.01, J=4

(b) 3-skills case, slip=guess=0.2, J=4

(c) 3-skills case, slip=guess=0.01, J=8

(d) 3-skills case, slip=guess=0.3, J=8

(e) 4-skills case, slip=guess=0.01, J=5

(f) 4-skills case, slip=guess=0.2, J=5

Figure 3: Experiment 2: Configurations of different slip and guess parameters and number of skills, J.

the main finding is wide reaching and warrants further investigations. The support for designing Q-matrices that satisfy the identifiability condition by single-skill items is compelling in the experiments conducted with synthetic data. The results clearly show such matrices yield more accurate student skills assessment. In particular, they show that Q-matrices that contains items that span the whole range of potential combinations of skills tend to yield lower skills assessment than Q-matrices that simply repeat the pattern of single-skill items.

The finding that tests composed of single-skill items are better for skills assessment is somewhat counter-intuitive, as intuition suggests that a good test should also include items with combinations of skills. But intuition also suggests that items that involve combination of skills are more difficult, and it may not simply be because they involve more than one skill. It might be that solving items that combine different skills in a single problem is a new skill in itself. This conjecture is in fact probably familiar to a majority of educators, and the current work provides formal evidence to support it. And the immediate consequence is that Q-matrices, as we currently conceive them, fail to reflect that a task that combines skill involves a new skill.

Ideally, future work should be conducted with real data. However, given that we do not know the real Q-matrix that underlies real data, investigating the questions raised by the current study is non trivial. Meanwhile, further experiments with synthetic data can be considered with different choices on student profiles distribution, and different number of skills involved. Besides, the case where slip and guess are unknown should also be considered, which involves a different identifiability requirement (G. Xu & Zhang, 2015).

References

Barnes, T. (2010). Novel derivation and application of skill matrices: The Q-matrix method. *Handbook on educational data mining*, 159–172.

Beck, J. E., & Chang, K.-m. (2007). Identifiability: A fundamental problem of student modeling. In *International conference on user modeling* (pp. 137–146).

Casella, G., & Berger, R. L. (2002). *Statistical inference* (Vol. 2). Duxbury Pacific Grove, CA.

Chen, Y., Liu, J., Xu, G., & Ying, Z. (2015). Statistical analysis of Q-matrix based diagnostic classification models. *Journal of the American Statistical Association*, *110*(510), 850–866.

Chiu, C.-Y. (2013). Statistical refinement of the Q-matrix in cognitive diagnosis. *Applied Psychological Measurement*, *37*(8), 598–618.

Chiu, C.-Y., Douglas, J. A., & Li, X. (2009). Cluster analysis for cognitive diagnosis: Theory and applications. *Psychometrika*, *74*(4), 633.

Culpepper, S. A. (2015). Bayesian estimation of the dina model with gibbs sampling. *Journal of Educational and Behavioral Statistics*, *40*(5), 454-476. Retrieved from `http://www.hermanaguinis.com/pubs.html` doi: 10.3102/1076998615595403

De La Torre, J. (2009). Dina model and parameter estimation: A didactic. *Journal of Educational and Behavioral Statistics*, *34*(1), 115–130.

de la Torre, J., & Chiu, C.-Y. (2015). A general method of empirical Q-matrix validation. *Psychometrika*, 1–21.

Desmarais, M. C., & Naceur, R. (2013). A matrix factorization method for mapping items to skills and for enhancing expert-based q-matrices. In *Artificial intelligence in education* (pp. 441–450).

Desmarais, M. C., Xu, P., & Beheshti, B. (2015). Combining techniques to refine item to skills q-matrices with a partition tree. In *Educational data mining 2015.*

Doroudi, S., & Brunskill, E. (2017). *The misidentified identifiability problem of bayesian knowledge tracing.* International Conference on Educational Data Mining, EDM2017.

Liu, J., Xu, G., & Ying, Z. (2012). Data-driven learning of Q-matrix. *Applied psychological measurement*, *36*(7), 548–564.

Liu, J., Xu, G., & Ying, Z. (2013). Theory of the self-learning Q-matrix. *Bernoulli: official journal of the Bernoulli Society for Mathematical Statistics and Probability*, *19*(5A), 1790.

Montgomery, D. C. (2017). *Design and analysis of experiments.* John wiley & sons.

Qin, C., Zhang, L., Qiu, D., Huang, L., Geng, T., Jiang, H., ... Zhou, J. (2015). Model identification and Q-matrix incremental inference in cognitive diagnosis. *Knowledge-Based Systems*, *86*, 66–76.

Tatsuoka, K. K. (1983). Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of educational measurement*, *20*(4), 345–354.

van De Sande, B. (2013). Properties of the bayesian knowledge tracing model. *Journal of Educational Data Mining*, *5*(2), 1–10.

Xu, G., & Zhang, S. (2015). Identifiability of diagnostic classification models. *Psychometrika*, 1–25.

Xu, P., & Desmarais, M. (2016). Boosted decision tree for Q-matrix refinement. In *Edm* (pp. 551–555).

Yan, D., Almond, R., & Mislevy, R. (2004). A comparison of two models for cognitive diagnosis. *ETS Research Report Series*, *2004*(1).

What can we learn from college students' network transactions? Constructing useful features for student success prediction.

Ian Pytlarz
Purdue University
West Lafayette, IN
ipytlarz@purdue.edu

Shi Pu[*]
Purdue University
West Lafayette, IN
spu@purdue.edu

Monal Patel
Purdue University
West Lafayette, IN
patel@purdue.edu

Rajini Prabhu
Purdue University
West Lafayette, IN
rajini@purdue.edu

ABSTRACT

Identifying at-risk students at an early stage is a challenging task for colleges and universities. In this paper, we use students' on-campus network traffic volume to construct several useful features in predicting their first semester GPA. In particular, we build proxies for their attendance, class engagement, and out-of-class study hours based on their network traffic volume. We then test how much these network-based features can increase the performance of a model with only conventional features (e.g., demographics, high school GPA, standardized test scores, etc.). We labeled students as "above median" and "below median" students based on their first term GPA. Several machine learning models were then applied, ranging from logistic regression, SVM, and random forests, to AdaBoost. The result shows that the model with network-based features consistently outperforms the ones without, in terms of accuracy, f1 score, and AUC. Given that network activity data is readily available data in most colleges and universities, this study provides practical insights on how to build more powerful models to predict student success.

Keywords

Student success prediction, Engagement, Attendance, Study time, Network activity.

1. INTRODUCTION

Students' academic performance is of interest for important practical reasons. To start with, one's college GPA is related to an individual's labor market performance [9, 16] and future educational pursuits [3]. More importantly, studies have shown that academic performance, especially in the early stage, is a strong predictor of students' retention [1, 5, 11]. Therefore, it could be used to identify at-risk students.

Unfortunately, predicting students' early academic performance is a challenge, essentially because it is difficult to obtain informative data. In this study, we propose to use students' on-campus network traffic volume to infer their location and behaviors. Through the inferred location and behavior, we construct several features that have been shown to be related to students' academic success, namely, attendance, in-class engagement, and out-class study effort. We then demonstrate that including these features into predicting models will improve the model performance in all conventional performance metrics.

Specifically, our research questions are:

1. How accurate is the location inferred from students' network traffic?
2. How much gain could we obtain by incorporating students' network inferred behavior in predicting their academic success?

2. RELATED STUDIES

Empirical studies have accumulated considerable evidence on the effect of attendance, engagement, and study time on a student's academic performance. The most rigorous literature comes from the Economics discipline, where experimental or quasi-experimental designs were applied. To name a few, in a randomized experiment, Chen and Lin [6] found that attendance increases students' final exam course grade by 9.4% – 18%. In another field experiment, Marburger [14] showed that mandatory attendance policy improves exam performance through reducing absenteeism. Using an instrumental variable approach, Stinebrickner and Stinebrickner [18] showed that college students' study time has a positive impact on their first year grade. In another study, Andrietti and Velasco [2] used first difference to remove time-invariant confounding variables, such as ability, in the estimating of effects of study time. They also found that study time had a large impact on students' final grades in two econometrics courses. Credé, Roch, and Kieszczynka [7] conducted a recent meta-analysis on the effect of attendance on grades. They found that attendance has strong relationships with both course grades and GPA.

In correlational studies, the well-cited work by Kuh, Cruce, Shoup, Kinzie, and Gonyea [13] showed that the time spent studying per week and the engagement in educational purposeful activities like asking questions in class are positively correlated to a student's first-year GPA. In a recent literature review, Trowler [19] concluded that studies in engagement in general found it to be positively correlated to student learning.

Though significantly correlated with academic performance, a student's behavioral data is difficult to obtain. Recent effort usually relies on measuring individuals' interaction with the learning management system as a proxy for their study effort [4, 8, 12, 15, 17]. Such practice has value, especially for the courses that are pre-

[*] Shi Pu is the corresponding author.

dominantly online. However, when the interested population are students taking courses on a traditional campus and learning management systems are mainly used as a mean to disseminate lecture notes and collect homework, interaction with the learning management system is unlikely to be an informative proxy for study effort.

To our best knowledge, this paper is the first study to use network traffic to build meaningful features in predicting students' academic success. A few previous works have demonstrated the possibility of inferring students' attendance through smartphone GPS and WiFi connections [10, 20, 21]. In general, they use smartphones to track individuals' location and check if students appear to be in class when they should be. These studies shed light on an innovative approach to collect real-time students' attendance data. However, all of them involve installing a third-party software, which provides an extra roadblock for scaling. As we will demonstrate later, students' attendance can also be inferred from their on-campus network traffic. This approach utilizes the existing network data, thus is arguably more scalable.

3. DATA AND METHOD

The study utilizes the data collected for an advanced learning analytics endeavor at Purdue University, namely Academic Forecast[1]. The project built cutting-edge machine learning models for students' course performance and accumulative GPA. Academic Forecast intends to identify student behaviors that are positively correlated with their academic performance and to encourage students to increase such beneficial behaviors. Though utilizing a part of the data from Academic Forecast, the models we experiment in this study are not directly related to the ones implemented for Academic Forecast.

The study utilize students' individual-level administrative and network traffic data from Purdue University[2]. The sample included all first-time, full-time freshmen that entered the university in fall 2017, with 7555 students in total. The response variable of interest is a student's fall semester GPA. The response is coded as 1 if a student's GPA is larger than the median, 0 otherwise. Notice that the choice of median ensures that the label is balanced. The network traffic volume provides two pieces of important information about students: 1) a student's approximate location (the campus building name) when s/he is connected to the network, and 2) a student's network traffic volume during a time period.

The first research question concerns how accurate the network inferred location is. To validate the inferred location, we need some form of ground truth. Fortunately, as many first-year students live on campus in Purdue, we can safely assume that most students should be in their residential buildings during early morning hours. Thus, we can compare the network-inferred location with students' on-file residential buildings[3].

The second research question concerns the contribution of network-inferred behavior data to prediction models. The follow paragraphs will briefly cover the construction of the network-inferred behaviors.

A student i is considered attending a registered course j's session k if the student appears to be in the building where the session k is held during the class time. Then, the average attendance rate for student i in the first semester is inferred by averaging student i's attendance across sessions and across all courses:

$$attend_{ijk} = \begin{cases} 1, & if\ bld_{it} = bld_{jk}, t \in [jk\ start, jk\ end] \\ 0, & o.w \end{cases}$$

$$attend_i = \frac{1}{n}\sum_{j=1}^{n}(\frac{1}{m_j}\sum_{k=1}^{m_j} attend_{ijk})$$

Note that n is the total number of courses a student i has in the first term. m_j stands for the total number of sessions for course j. bld_{it} indicates a student i's campus building id at time t , and bld_{jk} indicates the campus building id for course j at session k. Essentially, $bld_{it} = bld_{jk}$ if and only if a student i shows up in the class building during the scheduled class time.

A student's out-class study time is approximated by the total time spent in buildings that are predominantly used for learning purposes (indicated by bld_{study} in the formula), for example, libraries and active learning centers. Out-class time is obtained by excluding the time when a registered course is taking place. Formally:

$$study_i = \sum t \times (bld_{it} == bld_{study})$$

Where t does not belong to any scheduled class time for student i. Note that $bld_{it} == bld_{study}$ if and only if a student i is in a "study related" building at none-class time t.

In-class engagement for a student i in course j session k is inferred by the network traffic volume a student has during that class session. The average in-class engagement during the first semester is again averaged across sessions and across all courses. Noting that the higher the traffic volume, the more likely that a student is dis-engaged[4] in the class:

$$eng_i = \frac{1}{n}\sum_{j=1}^{n}(\frac{1}{m_j}\sum_{k=1}^{m_j} eng_{ijk})$$

The network-inferred behaviors, along with a set of pre-college variables, were then fed to several common machine learning algorithms to predict if a student is going to score higher than the median. The common pre-college variables include high school GPA, high school quality, standardized test scores, gender, residency, race, etc. A 20-fold cross-validation is applied to

[1] Website: https://www.academicforecast.org

[2] The scope and procedure of this study strictly follow a proved IRB. All of the analysis of the data occurs within the existing Purdue data security infrastructure and guidelines controlling data utilized for campus daily operations. Data can only be accessed via a machine controlled by Purdue data security protocols.

[3] Students' locations after the late night and before early morning were never used in any of our predictive models, due to potential privacy concerns. However, for the purpose of validating the merit of network inferred index, we checked *at the aggregate level* if students' night locations agree with their residential buildings on the book. We did not further investigate which students' inferred location and theoretical location did not match.

[4] This is not necessarily true for classes that entail the use of internet.

estimate the model performance on unseen data. All models used pre-defined hyper-parameters to avoid being over-optimistic on performance estimation.

4. RESULT

To validate the accuracy of our network-inferred location, we choose an early Tuesday morning in September 2017 that is neither a public holiday nor a university holiday. Recall that students should be in their dormitory rooms at this time, thus their on-file residential building could be used as a ground truth to validate our network inferred location.

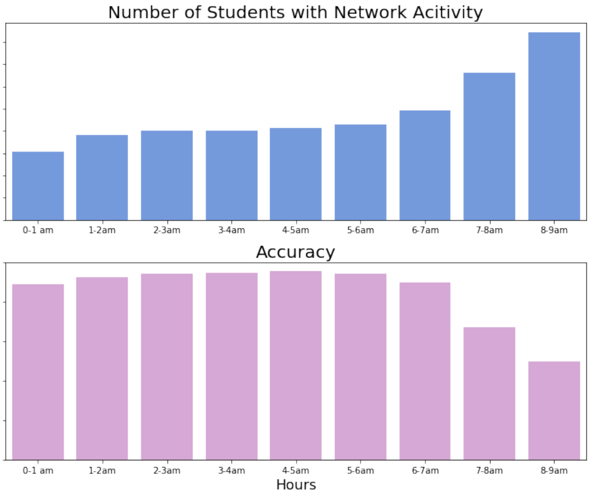


Figure 1. Validate network inferred location

Figure 1 demonstrate the result of this validation. As shown in the upper part of the graph, only a portion of all students living on campus have network activity before 6am, and the number increases rapidly after 7am. The lower part of the graph demonstrates the accuracy of the network inferred location. The accuracy is defined as the percentage of students whose on-file residential building agrees with the network inferred location. As we expected, the accuracy is high during the early morning, ranging from 89.20% - 95.70% between 0 to 6 am. The accuracy dropped rapidly after 7am; this plunge is likely due to the fact that students start to leave their residential buildings, thus it can no longer serve as a ground truth.

Students' on-file residential location never fully agrees with the network-inferred location. This does not necessary suggest that there is some noise in the inferred location, as we cannot be fully sure that all students are in their residential buildings at any given time. However, this result indicates that network-inferred location should be a good proxy for a student's real location. Thus, it can provide useful information on students' behaviors.

In Table1, we compare the classification accuracy between models with network features and the ones without. A 20-fold cross-validation is applied to estimate the model performance on unseen data. As the label is balanced (exactly 50% of students score higher than the median), accuracy serves a good performance metric. We experiment on several common algorithms to check if the performance gap is model dependent. All models used pre-defined hyper-parameters.

As shown in Table 1, models with network-inferred behaviors consistently outperform the models without network-inferred behaviors. The difference in accuracy ranges from 0.016 to 0.021. The right-most column records the t-statistics [5] for improved accuracy. The improvement is statistically significant at 0.05 level with one-side t-test for logistic model, random forest model, and AdaBoost model. The improvement on SVM model is only significant at 0.1 level. After including Bonferroni correction, only the improvement on AdaBoost remains statistically significant.

Table 1: Accuracy comparison : with/out network behaviors (t-test with Bonferroni correction, α = 0.05)

Classifier	No Network Behaviors	Network Behaviors	Diff	t-stat
Logistic	0.669 (0.02)	0.686 (0.03)	0.017	2.01*****
SVM	0.667 (0.03)	0.683 (0.03)	0.016	1.67
Random Forest	0.678 (0.03)	0.696 (0.03)	0.018	1.96*
AdaBoost	0.676 (0.03)	0.696 (0.03)	0.021	2.39*

Note: standard errors in parentheses

Table 2: Model performance comparison: with/out network behaviors

Classifier	Network	F1	Precision	Recall	AUC
Logistic	No	0.68	0.654	0.711	0.732
	Yes	**0.696**	**0.671**	**0.724**	**0.751**
SVM	No	0.672	0.661	0.687	0.726
	Yes	**0.683**	**0.678**	**0.691**	**0.747**
Random Forest	No	0.679	0.671	0.688	0.735
	Yes	**0.687**	**0.702**	**0.674**	**0.761**
AdaBoost	No	0.67	0.675	0.668	0.734
	Yes	**0.690**	**0.698**	**0.682**	**0.757**

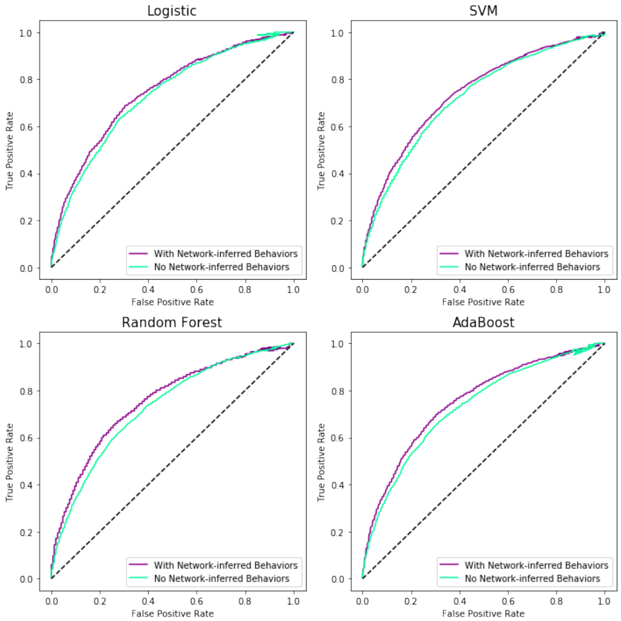


Figure 2. Mean ROC curves for different models, with v.s. with-out network-inferred behaviors

[5] Paired sample t-tests are used here to compare the difference between models with network features and the ones without.

Table 2 and Figure 2 report further performance comparison. Models with network-inferred behaviors again consistently outperform the models without such information on F1 score, precision, recall, and AUC. The differences are small but consistent in the ROC curves.

At last, Table 3 demonstrates the top 10 important feature importance in Random Forest and AdaBoost. Network-inferred behaviors are always among the top important features. In AdaBoost, student engagement is the single most important feature, followed by students' high school GPA, high school quality[6], study time, and attendance. The Random Forest relies disproportionally on students' high school GPA. Students' attendance, study time, and in-class engagement are more informative for the model than the rest predictors. Note that network-inferred behaviors are always more important than standardized test scores[7] in the two models.

Table 3: Feature importance[8]

Random Forest		AdaBoost	
Feature Name	Importance	Feature Name	Importance
High school GPA	0.385	**Engagement**	0.224
Attendance	0.157	High school GPA	0.145
Engagement	0.107	School quality	0.143
Study time	0.106	**Study time**	0.143
Std test score	0.091	**Attendance**	0.141
Zip code income	0.067	Zip code income	0.120
School quality	0.061	Std test score	0.059
Female	0.009	International	0.009
International	0.009	Asian	0.006
Asian	0.005	Hispanic	0.004

5. DISCUSSION

This study proposed a novel way to utilize on- campus network traffic data to improve student success prediction models. In particular, we have demonstrated that network-inferred location is a good proxy for students' actual location. Experimenting on a randomly chosen early morning, we found that 89.20% - 95.70% of students' network-inferred location matches their on-file residential location. In addition, we demonstrate that including the network traffic data improves the model performance in conventional performance metrics. Interestingly, the improvement is consistent across different models, ranging from the basic logistic regression models to more complicated ensemble classifiers.

The network-inferred behaviors are rooted in existing literature on student success. Namely, attendance, engagement, and study time have been found to be related to a student's GPA in various researches. Therefore, we believe the result should not be a peculiarity in Purdue's data but can be generalized to other colleges and universities.

In addition to generalization, our approach has two important practical advantages. First, models based on network-inferred behavior provide actionable suggestions for student advisors. To elaborate, the pre-college predictors can only tell advisors whether a student is well-prepared for college. Other analytical models usually only inform the advisor how well a student is doing in each class. Neither type of predictor could provide suggestions on *why* a student is having trouble. The network-inferred behaviors, on the contrary, could possibly pinpoint the student's action that directly leads to their poor performance, e.g., poor attendance. Second, network-inferred behaviors are based on existing network data in each university, thus the scaling cost is arguably low.

The study, nevertheless, has several important limitations. First, the chosen response is median GPA instead of more meaningful classifications, e.g., retention and academic probation. Therefore, it is unclear if the network features are still informative for detecting at-risk students. Second, the improvement in accuracy is limited. Future study should seek to uncover deeper pattern from the location data to improve model performance.

6. REFERENCES

[1] Allen, J. et al. 2008. Third-year college retention and transfer: Effects of academic performance, motivation, and social connectedness. *Research in Higher Education*. 49, 7 (2008), 647–664. DOI:https://doi.org/10.1007/s11162-008-9098-3.

[2] Andrietti, V. and Velasco, C. 2015. Lecture Attendance, Study Time, and Academic Performance: A Panel Data Study. Journal of Economic Education. 46, 3 (2015), 239–259. DOI:https://doi.org/10.1080/00220485.2015.1040182.

[3] Astin, A.W. 1993. What matters in college? Liberal Education.

[4] Brinton, C.G. and Chiang, M. 2015. MOOC performance prediction via clickstream data and social learning networks. *2015 IEEE Conference on Computer Communications (INFOCOM)* (2015), 2299–2307.

[5] Cabrera, A.F. et al. 1993. College Persistence: Structural Equations Modeling Test of an Integrated Model of Student Retention. *The Journal of Higher Education*. 64, 2 (1993), 123. DOI:https://doi.org/10.2307/2960026.

[6] Chen, J. and Lin, T.F. 2008. Class attendance and exam performance: A randomized experiment. *Journal of Economic Education*. 39, 3 (2008), 213–227. DOI:https://doi.org/10.3200/JECE.39.3.213-227.

[7] Crede, M. et al. 2010. Class Attendance in College: A Meta-Analytic Review of the Relationship of Class Attendance With Grades and Student Characteristics. *Review of Educational Research*. (2010). DOI:https://doi.org/10.3102/0034654310362998.

[8] Jiang, S. et al. 2014. Predicting MOOC Performance with Week 1 Behavior. *Proceedings of the 7th International Conference on Educational Data Mining (EDM)*. (2014), 273–275.

[9] Jones, E.B. and Jackson, J.D. 1990. College Grades and Labor Market Rewards. *Journal of Human Resources*. 25, 2 (1990), 253–266. DOI:https://doi.org/10.2307/145756.

[10] Kassarnig, V. et al. 2017. Class attendance, peer similarity,

[6] Measured by the average Purdue GPA for students come from that high school.

[7] Constructed based on SAT and ACT scores.

[8] A feature's importance in Random Forest is the average decrease in impurity by that feature across all trees, the higher the better. AdaBoost's feature importance depends on the base learner, which are decision trees in this study. Therefore, its feature importance is again calculated by averaging the decrease in impurity by each feature.

and academic performance in a large field study. *PLoS ONE*. (2017). DOI:https://doi.org/10.1371/journal.pone.0187078.

[11] Kern, C. et al. 1998. Correlates of College Retention and GPA- Learning and Study Strategies, Testwiseness, Attitudes and ACT. *Journal of College Counseling*.

[12] Kloft, M. et al. 2014. Predicting MOOC Dropout over Weeks Using Machine Learning Methods. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. (2014), 60–65.

[13] Kuh, G.D. et al. 2008. Unmasking the Effects of Student Engagement on First-Year College Grades and Persistence. *The Journal of Higher Education*. 79, 5 (2008), 540–563. DOI:https://doi.org/10.1353/jhe.0.0019.

[14] Marburger, D.R. 2006. Does Mandatory Attendance Improve Student Performance? *The Journal of Economic Education*. 37, 2 (2006), 148–155. DOI:https://doi.org/10.3200/JECE.37.2.148-155.

[15] Phan, T. et al. 2016. Computers & Education Students ' patterns of engagement and course performance in a Massive Open Online Course. *Computers & Education*. 95, (2016), 36–44. DOI:https://doi.org/10.1016/j.compedu.2015.11.015.

[17] Sinha, T. et al. 2014. Your click decides your fate: Inferring Information Processing and Attrition Behavior from MOOC Video Clickstream Interactions. (2014).

[18] Stinebrickner, R. and Stinebrickner, T.R. 2008. THE CAUSAL EFFECT OF STUDYING ON ACADEMIC PERFORMANCE. The BE Journal of Economic Analysis & Policy. 8, 1 (2008), 14. DOI:https://doi.org/10.1017/CBO9781107415324.004.

[19] Trowler, V. 2010. Student engagement literature review. *Higher Education*. (2010), 1–15.

[20] Wang, R. et al. StudentLife: Assessing Mental Health, Academic Performance and Behavioral Trends of College Students using Smartphones. DOI:https://doi.org/10.1145/2632048.2632054.

[21] Zhou, M. et al. EDUM: Classroom Education Measurements via Large-scale WiFi Networks. DOI:https://doi.org/10.1145/2971648.2971657.

Mining Student Misconceptions from Pre- and Post-Test Data

Ángel Pérez-Lemonche
Universidad Autonoma de Madrid
Ciudad Universitaria de Cantoblanco,
28049 Madrid, Spain
angel.perezl@estudiante.uam.es

Byron Coffin Drury
MIT
77 Massachusetts Avenue
Cambridge, MA 02139, USA
bdrury@mit.edu

David Pritchard
MIT
77 Massachusetts Avenue
Cambridge, MA 02139, USA
dpritch@mit.edu

ABSTRACT

We analyze results from paired pre- and post-instruction administration of the Mechanics Baseline Test to 2238 students in introductory mechanics classes. We investigate pairs of specific wrong answers given with unusual frequency by students on the pretest. We also identify transitions between pre- and post-test answers on the same question which elucidate student learning due to instruction. We define criteria for excess transitions above a random response model. Some common transitions are found to be associated specifically with students within a particular range of skills. Further, transitions from pre- to post-test revealed that incorrect pretest answers that were frequently repeated on the post-test often correspond to known misconceptions from physics or math. Thus, our data mining techniques can elucidate common student misunderstandings of mechanics concepts and how instruction affects these misunderstandings. This opens the way for finding improved interventions for specific misunderstandings revealed by analyzing results from pre- and post conceptual tests.

Keywords

Pre- and post- Testing; Common Student Misconceptions; Educational Data Mining; Analyzing Wrong Answers.

1. INTRODUCTION

The Force Concept Inventory [9] by Hestenes group revolutionized physics instruction by showing that students trained mostly on end-of-chapter problems in standard textbooks did not learn to answer easy (so teachers thought) questions based on fundamental concepts in the domain. This has led to tremendous reform of physics instruction worldwide and a series of concept tests covering introductory physics and astronomy [7]. The present study uses another research-based assessment, the Mechanics Baseline Test ("MBT"). The MBT is designed for students with more physics background and is appropriate for introductory students at MIT.

Research-based assessments such as concept inventories and surveys are typically developed by first administering the questions in open response format. Analysis often reveals clusters of related responses which are then made into distractors in a multiple-choice version of the assessment. Since these assessments typically center only a particular subdomain, e.g. force and motion, a part of Newtonian mechanics, it is expected that common misconceptions (also called alternate conceptions and misunderstandings) will manifest as correlated selections of distractors to different questions. We searched for these, as well as for statistically significant deviations of specific learning transitions from a random guessing hypothesis.

This paper addresses several questions relative to the deep assessment of students' knowledge structure based on results on the Mechanics Baseline Test. Our objective is to find the 'atomic' student conceptions and abilities that underlie their answers to the questions (possibly incorrectly)? Our approach is data mining on a large sample of pre and post-tests, and concentrates on these research questions

- Are there pairs of wrong answers to different questions that reveal common misunderstandings?
- Are there exceptionally prevalent transitions from pre- to post-test that seem to indicate learning some specific knowledge?
- Can we suggest new questions or improvements to existing ones that will improve the assessment?

We are not the first to attempt to extract actionable analysis from concept tests. Indeed, the FCI has been analyzed using factor analysis [4]; however, that analysis has been questioned [5]. The MBT has been refined using Item Response Theory analysis [3]. Recently Brewe et al. [2] have applied Network analysis to the FCI to predict post scores. The Colorado Learning Attitudes about Science Survey [1] has a nice web-based multicategory analysis based on factor analysis that is used. But it's fair to say that most concept tests are not analyzed beyond the score and whether it seems appropriate for each particular class based on quality of students & instructional style [8]. This provides a good characterization of the students' (and class) overall knowledge and gives a useful indication of the amount of learning if the assessment is administered both pre- and post-instruction. Unfortunately, such one-dimensional analysis ignores the category-specific information that the method of construction of these assessments would seem to generate. Therefore, administering these assessments neither informs the student about which concept(s) they know well or poorly nor informs the teacher about the areas in which they most need to improve their instruction.

The goal of finding specific difficulties and misconceptions of students continues to appear reasonable yet remains tantalizingly out of reach. The progress made here shows the promise of analysis of learning data at scale. But while our findings are clearly revelatory, they beg for further development to make them useful. We discuss ways of closing this gap in the last section: Future.

Table 1: The students in our dataset represent five years of an introductory mechanics course at MIT. Since some students lack either a pre- or post-test score, we have calculated grades and normalized gain using only those students who took both tests. The pretest was administered at the beginning of the semester, and the post-test was administered – often as part of the final exam - at the end of the semester.

year	#pre	#post	#both	fraction pre	fraction post	gain
2005	485	509	438	.57±.15	.66±.13	0.34
2007	356	356	355	.56±.15	.76±.12	0.46
2008	414	414	410	.58±.15	.79±.12	0.51
2009	612	565	527	.58±.14	.75±.12	0.41
2010	589	554	508	.60±.18	.78±.12	0.44
all	2456	2398	2238	.58±.15	.75±.15	0.40

2. CORRELATIONS ON PRE- AND POST-TESTS

Assuming that there are fundamental misconceptions shared by many students, the question becomes "how can we detect these in the test results". Since the MBT was designed with distractors compiled from open responses to those questions, one would expect that a specific misconception would lead students to give a specific wrong answer. If a misconception leads to wrong answers on two (or more) questions, we expect that students with this misconception would submit this particular pair of wrong answers with more than random frequency. We seek to detect such correlated pairs of wrong answers by looking for statistically excessive pairs of wrong answers, and that these will offer insight into the nature and prevalence of specific student misconceptions. We examine only correlations between wrong answers, since correct answers do not provide much information about misconceptions.

Table 2: Correlations between wrong answers on MBT pretest. For each pair of correlated wrong answers we show the overall correlation coefficient, the fraction of all students who gave the paired response, the Student's t-statistic, and the p-value. X indicates that a student did not answer the question (this is considered as a specific response).

Responses 1&2	Correlation [%]	Fraction [%]	t	p-value
Q1A Q2E	67	13	15.1	$\sim10^{-37}$
Q4D Q5C	41	19	9.3	$\sim10^{-18}$
Q11X Q12X	57	7	8.5	$\sim10^{-14}$
Q9X Q11X	48	8	7.4	$\sim10^{-11}$
Q9X Q12X	47	7	6.8	$\sim10^{-10}$
Q13A Q14A	52	5	6.0	$\sim10^{-8}$
Q13X Q14X	60	2	4.7	$\sim10^{-5}$
Q20B Q21C	35	7	4.6	$\sim10^{-5}$
Q20D Q22D	44	4	4.6	$\sim10^{-5}$
Q20A Q22C	19	15	3.5	0.001
Q16C Q16D	18	13	3.0	0.004

We examined all possible wrong answer pairs, defining a binary variable for each possible wrong answer, specifying whether a particular student did or did not give that answer. We calculated the tetrachoric correlation between every pair of answers, as well as the amount by which the observed number of students giving the paired wrong answers exceeded the number expected assuming that each wrong answer was selected independently at random with the observed answer probability distribution for each question alone. All pretest correlations found to be significant at the p = 0.01 level are displayed in Table 2.

Because students with very low skill may have weak or inconsistent preconceptions and students with very high skill presumably have few misconceptions of any sort, we expect that certain misconceptions will be held primarily by students lying within a limited range of overall ability or perhaps in students only of low ability. To test this hypothesis, we divided the students into 7 equal partitions sorted by overall score and calculated correlation coefficients for each partition independently.

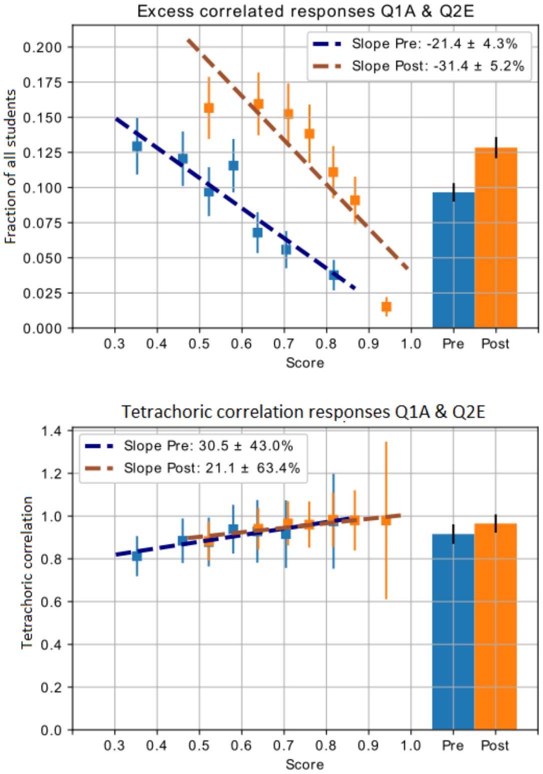


Figure 1: Questions 1 & 2: Velocity and Acceleration Graphs, correlation of 1A and 2E. The tetrachoric correlation and excess paired responses are plotted in each of seven cohorts divided by overall score.

In Questions 1 and 2, shown in Figure 1, the paired errors both correspond the same misinterpretation of a stroboscopic image of an accelerating object. The very high correlation coefficient implies that roughly 90% of the students who answered 1A also answered 2E. This suggests that the students determined the acceleration (Q2) from the answer to the velocity (Q1), thereby making the same time-base error. This hypothesis is supported by the fact that the better cohorts made relatively fewer mistakes carrying out this prescription, hence had (even) higher correlations, as did all students on the post-test.

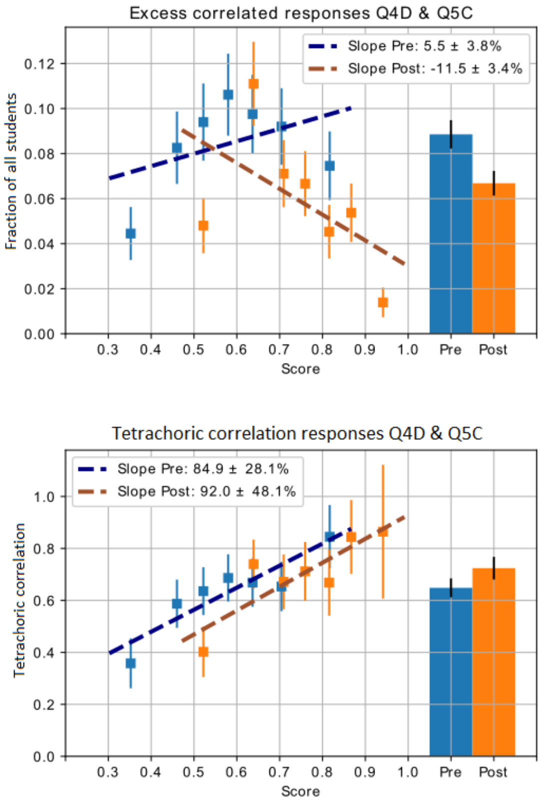
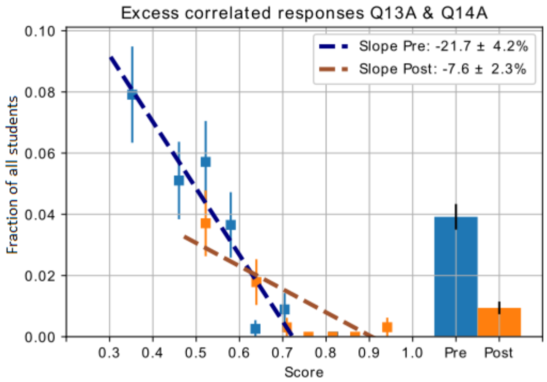


Figure 2: Question 4 and 5: Direction of Acceleration on Ramp - correlation of 4D with 5C.

Correlated wrong answers on Questions 4 and 5, shown in Figure 2, both correspond to ignoring real forces when applying F=ma. It is apparent that the prevalence of this error maximizes at score levels ~ 0.6 suggesting a specific misconception that shows some, but not too much, knowledge.

Correlated responses 13A and 14A both correspond to confusing the mass of a system with the force required to support it. This correlation is very strong (R ~ 0.9), but the probability of making this error drops dramatically with score, reflective of the fact that the associated error is virtually at a random rate with prevalence < ½% for all students scoring above 75% (where the correlation has huge errors). This seems to be an error predominantly made by low-ability students, and we suggest that it results from omitting g=10 m/s^2 when calculating weight from mass.



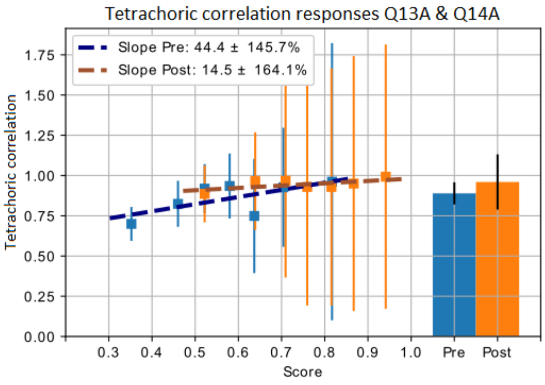


Figure 3: Questions 13 & 14: Elevator with Two Hanging Blocks – correlations between 13A and 14A.

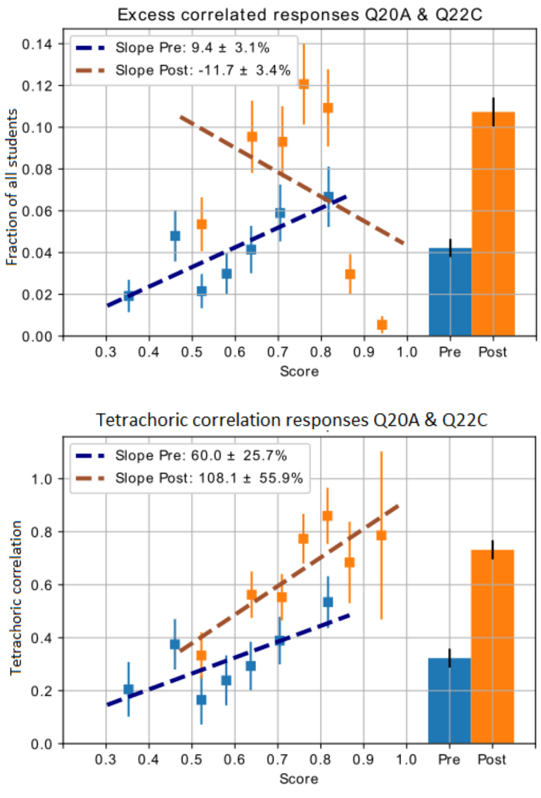


Figure 4: Questions 20 & 22: Pushing Different Masses the Same Distance with the Same Force, correlation of 20A & 22C.

The triplet of questions 20 through 22, Pushing Different Masses the Same Distance with the Same Force, yields several highly correlated pairs of wrong answers. These problems, particularly 20 and 22, are among the most difficult on the test, with respectively 36% and 47% of students answering them correctly on the pretest.

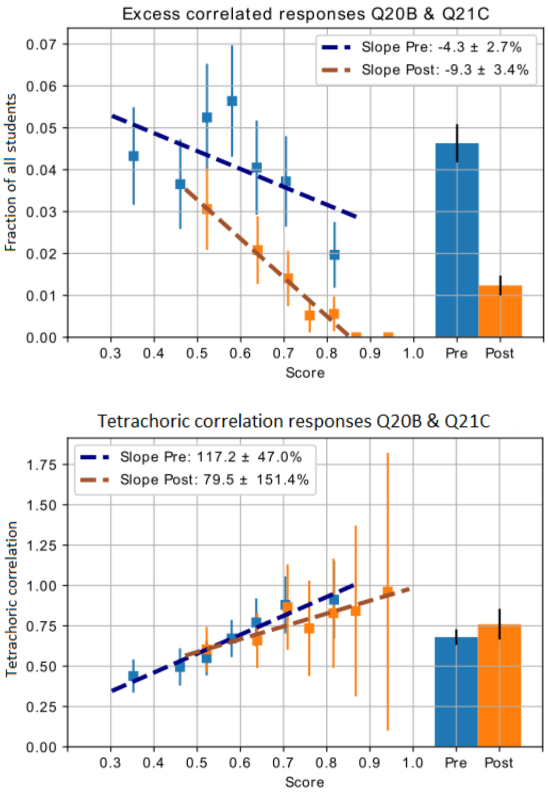


Figure 5: Questions 20 & 21: Pushing Different Masses the Same Distance with the Same Force, correlations between 20B and 21C.

The correlated pair consisting of 20A and 22C correspond to confusing the change in energy of a system with the change in momentum. About 4% of students showed this excess pairing on the pretest, rising to 10% on the post-test, the most dramatic of the only two increases in excess correlated responses found in this study. There is clear evidence that this excess correlation has a peak, probably around score 75%. Together with the dramatic increase in excess correlated responses on the post, we argue that this paired response requires confusion of work with impulse augmented by some understanding of momentum.

Similarly, the responses 20B and 21C, shown in Figure 5, seem to correspond to the idea that equal force results in equal acceleration, regardless of mass. This response decreases with increasing score and also from pretest to post-test. The correlation coefficient increases dramatically with score on both pre- and post-test.

The final correlated pair that comes from questions 20-22 is 20D and 22D, shown in Figure 6. These answers are both "too little information" to calculate the energy and momentum of two pushed pucks. Not surprisingly, this paired response shows the greatest decrease from pre- to post-test (~ 5:1), presumably because most students learn about either energy or momentum during the course.

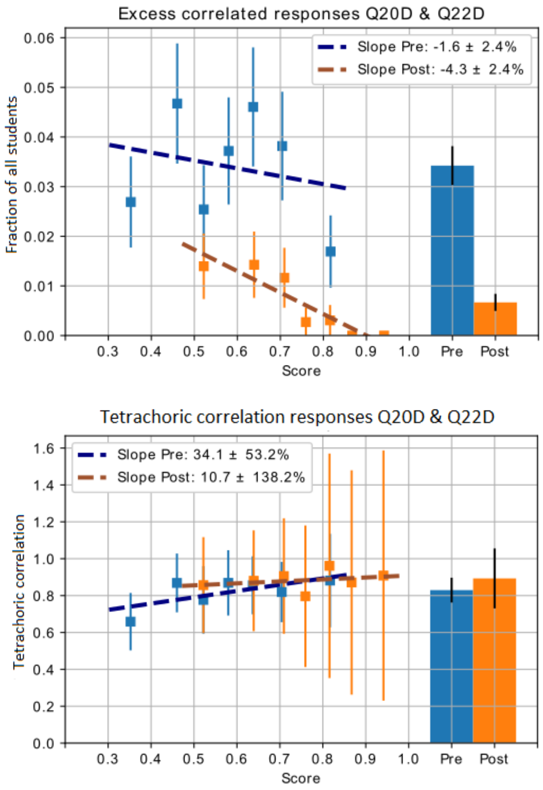


Figure 6: Questions 20 & 22: Pushing Different Masses the Same Distance with the Same Force: correlation between 20D and 22D.

3. TRANSITION ANALYSIS: PRE →POST ON THE SAME QUESTION

3.1 Robust Wrong Answers: Null Hypothesis and Findings

If a certain wrong answer on the pretest corresponds to an entrenched misconception, students should give that same answer on the post-test. We therefore use a baseline null hypothesis for comparison that assumes that students answer the post-test independently of their response on the pretest. We search for "excess" transitions above this null. When looking for wrong answers which are unusually strongly held (what we call "robust wrongs"), for example, our null hypothesis is that the student is unaffected by instruction and would answer with the same probability on the post-test as on the pretest. If 30% of all students answered correctly on the pretest, this would imply a 9% robust rate. This null hypothesis would reflect reality if all students were guessing on both pre and post.

The most robust wrong answer seen in Table 3, answer E on question 12, is "none of the above" on a numerical question, which does not suggest a specific physics misconception. The next two correspond to the same error in interpreting the motion diagram in a related pair of questions, namely reversal of the time axis. The fourth corresponds to claiming that the middle of the range of a graphed function is its average value. The fifth indicates that students have erroneously used the mass of part of a system instead of the total mass of the system in F=ma, and the sixth involves treating the speed of an object as its acceleration in an F=ma problem. The first four of these give little insight into physics misconceptions, though they do seem to highlight mathematical deficiencies, but the robust wrong responses in Q17 and Q13 reveal difficulty with applying Newton's Second Law. Confusing speed and acceleration is a well-known student misconception.

Table 3: Six wrong answers were given by students on both the pre- and the post test at rates which were significantly greater than chance at the p<0.0001 level. Here we present the p-values for each of these responses and the frequency with which these responses were given as a percentage of all responses to the questions.

p-value	Question	%
~10^{-9}	Q12E	14
~10^{-8}	Q1A	6
~10^{-7}	Q2E	5
~10^{-6}	Q25B	4
~10^{-5}	Q17C	10
~10^{-4}	Q13C	3

3.2 Wrong to Correct: Null Hypothesis and Findings

Since the wrong answers on the MBT are designed to represent specific misconceptions, the question arises of whether students who give certain wrong answers on the pretest might be more or less likely than other students to subsequently provide the correct answer on the post-test. In other words, we wish to ascertain whether some misconceptions are more resistant to instruction than others. In calculating the excess (or deficit) relative to chance of students making a transition from a wrong answer to the correct answer, our null hypothesis is again that a student's likelihood of answering correctly on the post-test is independent of the answer they gave on the pretest. However, we must take into account that a non-trivial fraction of the students answer any given problem correctly on both pre- and post-test not by chance but because they understand the relevant physical concepts-- in some cases as many as 80% of students answered a problem correctly on both tests. We therefore use a slightly different null hypothesis that eliminates students who do not change their answer after instruction. This posits that the conditional probability of a student offering the correct answer to a particular problem given that they gave a particular incorrect answer to that problem on the pretest should be equal to the ratio of the number of students who transitioned to the correct answer from any incorrect answer over the total number of students who changed their answer in any direction. The most statistically significant wrong to correct transitions are displayed in Table 4 and discussed below.

3.2.1 Q1 and Q2: Find velocity and acceleration from a graph

Both transitions have moderate excess probability (~ 60%) of switching to the correct answer, and very small probability that the wrong is robust. This suggests that these wrongs are mainly due to careless errors in reading the graph.

3.2.2 Q14: Force from lower rope on top block of two hanging in stationary elevator

About 6.5% answered D (20N, twice the answer) or A (forgot multiplying by g) and at least 80% of both switch to correct. This generally shows strong growth on applying Newton's Laws. (Although most students probably saw this example in the course.) The very small number of robust wrongs shows that the initial answers may have been mostly due to lack of full understanding of tension rather than strongly held misconceptions.

3.2.3 Q23: Average acceleration from graph of velocity versus time

The two most attractive wrong answers, taking v=0 at t=0 (p < 10-4) and "none of above" ($p < 10^{-3}$) both exhibited excess transitions to the correct answer. Students with pre-answers switched to correct with 78% and 80% likelihood. This is a graphing question, so possibly learning about graphs is reinforced due to complimentary instruction on graphs of functions in the introductory calculus courses which a majority of students are co-registered for. NOTE: 14% of those who were correct on the pretest answered incorrectly on the post.

Table 4: Wrong to correct transitions which occur significantly more frequently than would be expected due to chance. We display the p-values and the overall frequency with which the transition occurred for all such transitions with p<0.001.

p-value	Transition	Freq. [%]
~10^{-6}	Q2E2D	10
~10^{-5}	Q1A2B	11
~10^{-4}	Q23C2D	9
~10^{-3}	Q14D2B	8

4. CONCLUSIONS AND DISCUSSION

4.1 Excess Correlated Wrong Responses

"Excess correlated responses" (ECR) are in addition to those that would occur if the correlated questions were independently answered randomly with the observed frequency of wrong answers. Correlated wrong answers between different questions were detected and described in two ways: by the excess fraction of students who selected both wrong answers (vs. assuming independently answered questions), and by the fraction of students who selected one wrong answer who also selected the other (tetrachoric correlation). Both quantities varied considerably with the overall ability of the students as measured by their overall fraction correct (score) on the assessment. For this reason, we discuss only results specific to student overall score.

The correlated wrong answers found here are surprisingly prevalent, with ~10% or more of the students in one of the score groups selecting both of the paired wrongs in all cases except the last two which have the lowest statistical significance. Our most important findings are:

1. The percentage of correlated wrongs always drops for students with score >0.7, and typically decreases to 1% or lower for the top score group on the post-test.
2. In the two cases suggesting a real misconception, force from ramp and kinetic energy of masses, the percentage of correlated wrongs also decreased for the lowest-scoring groups.

The tetrachoric correlation measures the "purity" of the observed correlations. In every case presented, it reaches or exceeds 0.8 for groups with high test scores. This shows that essentially every skilled student giving one of the paired wrong answers also gives the other. Equivalently, the mistake or misconception is the main cause of the wrong answers on both questions. In cases where low-skill students appear to lack the correct physics knowledge (energy/momentum and direction of force on curved ramp), the correlation decreases to well below 0.5. Low tetrachoric correlation probably indicates that students are using a variety of incorrect reasons in their responses, so that many are led to answer one of the paired wrong answers but not the other.

In summary, the search for excess correlated answers has revealed two cases where the excess peaks for students in a particular range of overall score. This is a clear guide for instruction: if you teach a class is in this score range, then you should carefully address situations like this to tease out and rectify the underlying misconception. Additionally, the dramatic increase in mistaking work as the source of momentum on the post-test indicates that our instruction has to be clarified on this point. We find that the correlation of all wrong answer pairs increases for better students - indicating that this misconception is the main reason for these wrong answers and is being consistently applied to both questions I.e. skillful students don't make errors on just one of the problems due to some reason unrelated to the identified misconception.

4.2 Excess and Robust Transitions

We found that the none of the transitions from wrong to right indicated that that particular wrong answer was conceptually closely related to the correct answer; rather it seemed that the wrongs were due to careless responses or fuzzy thinking. On the other hand, several of the robust wrong answers seemed to reflect physics misconceptions.

5. SUMMARY

The probability of each particular ECR varies substantially with the overall ability (measured by total score) of the students, ranging up to a maximum of 4,5,8,10, and 11%. Although it always drops to ~ 1% or less at the highest ability, we find examples where the probability of ECR peaks at low and at medium student score. In all cases, the fraction of students giving one wrong answer who also give the other exceeds 80% for the highest-scoring students. This suggests that teachers concentrate on remediating ECR's common to their students' scores. ECR's seem to be a good method to detect significant misconceptions or missing knowledge held by students of a particular ability.

The transition analysis showed that robust wrongs often reflected misconceptions in math or physics, but that excess transitions from wrong to correct generally reflected carelessness rather than a mindset primed for learning the correct response.

6. FUTURE DIRECTIONS

The present work offers a new method for finding excess correlations of wrong answers between different questions, and particularly common (or uncommon) learning transitions within one question from pre- to post-test. Two future directions seem important to explore:

1. This method should be compared with network analysis which has a similar objective [2].
2. The students at MIT are significantly stronger than most who take the MBT. It is therefore important to extend the analysis to students with lower overall ability as evidenced by lower overall scores on the pre-test.
3. We have a new way to assess misconceptions; this should enable us to find better ways to remediate them.

7. ACKNOWLEDGEMENTS

We thank the Office of Digital Learning at MIT for financial and technical support. ÁPL thanks the Distinguished Scholar program of Spain for support.

8. REFERENCES

[1] Adams, W., Perkins, K., Podolefsky, N., Dubson, M., Finkelstein, N., & Wieman, C. (2006). New instrument for measuring student beliefs about physics and learning physics: The Colorado Learning Attitudes about Science Survey. *Physical Review Special Topics - Physics Education Research*, *2*(1), 10101. http://doi.org/10.1103/PhysRevSTPER.2.010101

[2] Brewe, E., Bruun, J., & Bearden, I. G. (2016). Using module analysis for multiple choice responses: A new method applied to Force Concept Inventory data. *Physical Review Physics Education Research*, *12*(2), 1–19. http://doi.org/10.1103/PhysRevPhysEducRes.12.020131

[3] Cardamone, C. N., Abbott, J. E., Rayyan, S., Seaton, D. T., Pawl, A., & Pritchard, D. E. (2011). Item response theory analysis of the mechanics baseline test. In *Physics Education Research Conference* (Vol. 1413, pp. 135–138). Omaha, Nebraska. Retrieved from http://dspace.mit.edu/handle/1721.1/78319

[4] Heller, P., & Huffman, D. (1995). Interpreting the force concept inventory: A reply to Hestenes and Halloun. *The Physics Teacher*, *33*(8), 503. http://doi.org/10.1119/1.2344279

[5] Hestenes, B. D., & Halloun, I. (1995). Interpreting the Force Concept Inventory A response to Huffman and Heller. *The Physics Teacher*, 502–506.

[6] Hestenes, B. D., & Wells, M. (1992). A Mechanics Baseline Test, (March).

[7] Lindell, R. S., Peak, E., & Foster, T. M. (2007). Are they all created equal? A comparison of different concept inventory development methodologies. *Physics Education Research Conference*, *883*, 14–17. Retrieved from http://scitation.aip.org/content/aip/proceeding/aipcp/10.1063/1.2508680%5Cnpapers3://publication/doi/10.1063/1.2508680

[8] McKagan, S. (2018). Physport.

[9] Swackhamer, G., Hestenes, D., & Wells, M. (1992). Force concept inventory. *The Physics Teacher*. http://doi.org/10.1119/1.2343497

Predicting Learning by Analyzing Eye-Gaze Data of Reading Behavior

Ramkumar Rajendran
Vanderbilt University
Nashville, TN, USA
ramkumar.rajendran@vanderbilt.edu

Anurag Kumar
Vanderbilt University
Nashville, TN, USA
anurag.kumar@vanderbilt.edu

Kelly E. Carter
Peabody College
Vanderbilt University
Nashville, TN, USA
kelly.e.carter@vanderbilt.edu

Daniel T. Levin
Peabody College
Vanderbilt University
Nashville, TN, USA
daniel.t.levin@vanderbilt.edu

Gautam Biswas
Vanderbilt University
Nashville, TN, USA
gautam.biswas@vanderbilt.edu

ABSTRACT

Researchers have highlighted how tracking learners' eye-gaze can reveal their reading behaviors and strategies, and this provides a framework for developing personalized feedback to improve learning and problem solving skills. In this paper, we describe analyses of eye-gaze data collected from 16 middle school students who worked with Betty's Brain, an open-ended learning environment, where students learn science by building causal models to teach a virtual agent. Our goal was to test whether newly available consumer-level eye trackers could provide the data that would allow us to probe further into the relations between students' reading of hypertext resources and building of graphical causal maps. We collected substantial amounts of gaze data and then constructed classifier models to predict whether students would be successful in constructing correct causal links. These models predicted correct map-building actions with an accuracy of 80% (F1 = 0.82; Cohen's kappa κ = 0.62). The proportions of correct link additions are in turn directly related to learners' performance in Betty's Brain. Therefore, students' gaze patterns when reading the resources may be good indicators of their overall performance. These findings can be used to support the development of a real-time eye gaze analysis system, which can detect students reading patterns, and when necessary provide support to help them become better readers.

Keywords

Eye-Gaze Data Analysis; Computer-Based Learning Environment; Reading Behavior; Classification.

1. INTRODUCTION

In a number of computer-based learning environments (CBLEs), students are expected to learn and refresh their domain knowledge from resources (typically in text or hypertext form with figures), then to construct solutions to assigned problems based on their learned knowledge. Such environments are known to help students develop cognitive skills and strategic reasoning processes, and, therefore, help students not only learn the domain content but prepare them for future learning [2, 3, 5, 17, 30-32]. However, because of the open-ended nature of these environments, novice learners often have difficulties in making progress toward their goals and completing their solutions. Therefore, the ability to track and understand learners' performance and behaviors is important for their overall success, so that relevant personalized feedback and instruction can be provided to them as necessary. However, tracking students' reading behaviors with sufficient precision and accuracy in computer-based learning environments is a non-trivial task.

Use of technologies, such as eye tracking devices can provide behavioral metrics that researchers can use to study learners basic cognitive processes and other information processing skills during reading [12, 27, 28, 35]. For educational research and applications, use of eye-tracking data has mainly focused on studying the effects of instructional strategies on eye-gaze behavior [21]. Some of these studies focus on learning how students' spatial contiguity [16], attention level [23] and viewing behavior [1] affect the cognitive processes that mediate learning outcomes. Conati et al. [7] have reviewed previous studies that modeled students' cognitive, metacognitive and affective states in intelligent learning environments using eye-gaze data. For example, Bondareva, et al. [4] assessed student learning from eye-gaze data during interaction with MetaTutor, an intelligent CBLE designed to develop self-regulated learning skills when generating summaries after reading about complex science topics. The MetaTutor study reported 78% classification accuracy on student learning based on the features extracted by gaze data alone. Similar results were reported by Kardan and Conati [18], in modeling students' learning with interactive simulations.

Peterson, et al. [25] report that learners' eye-gaze and pupil dilation data were used to predict performance and learning gains in Chem-Tutor, designed to teach chemistry. Hutt, et al. [14] studied students' mind wandering using eye-gaze on specific areas of interest (AOI) [10]. All of these results show that eye-tracking devices help to track learners' reading behaviors in CBLEs. Most of this research has relied upon expensive research-grade eye-tracking devices appropriate primarily for lab settings. However, newly available consumer-level eye-trackers are relatively inexpensive and have recently been deployed in classroom environments [14]. Our goal in this study is to run an initial proof of concept case study to demonstrate that these consumer-grade eye-tracking devices with sampling rates less than 90 Hz can effectively predict learners' behaviors in CBLEs.

In the research reviewed above [1, 4, 7, 16, 18, 22, 23], eye gaze features were extracted using global gaze features computed across broad Areas of Interest (AOI) that do not differentiate between more fine-grained screen contents. For example, the features extracted in [4] are based on predefined window position in the learning environment. This can be a limiting factor in CBLEs, where students are expected to learn by combining information from multiple hypertext resources. In Betty's Brain, a CBLE developed by our group [3, 24], students build a causal map to teach their agent, using hypertext resources that span multiple pages. Students are expected to find, read, and interpret sentences that provide information about entities and causal relations between entities, and add the link(s) to the current causal model. Extracting students' eye-gaze features as they read these hypertext resources would require a different AOI for each hypertext resource page. To address this challenge we propose a methodology to extract eye-gaze features that are directly related to content in each of the hypertext resource pages.

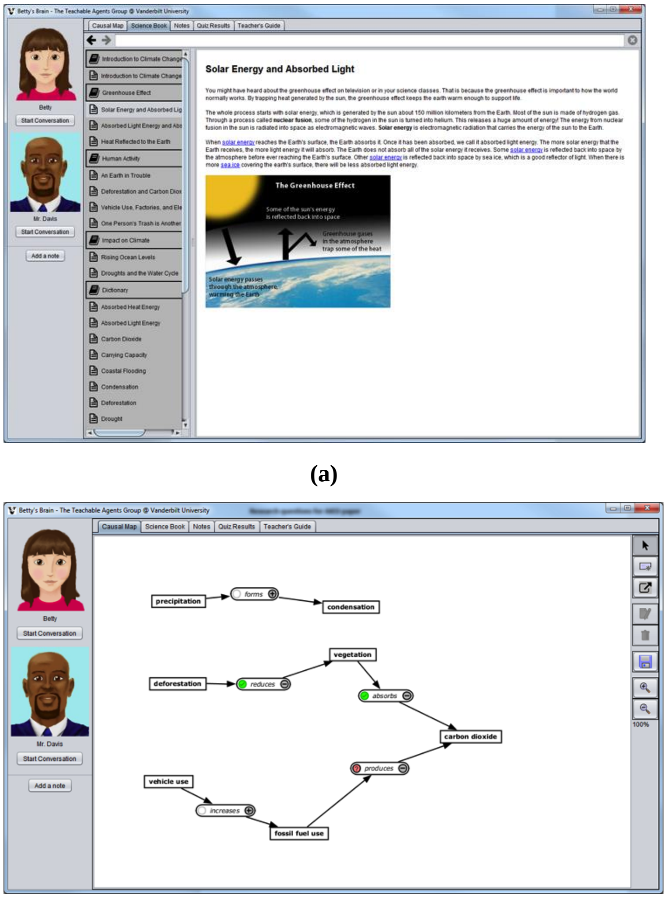
The proposed methodology was applied to eye-gaze data collected from middle school students who worked on Betty's Brain learning environment. The features extracted from the eye-gaze data were then used to construct classifier models that predict learners' model building effectiveness given their reading characteristics. For our study, we were able to predict learner performance in causal map building with an accuracy of 80% (F1 = 0.82; Cohen's kappa κ = 0.62). The learned classifier model was then used to classify learners reading behavior and directly related to learners' performance on map building action in Betty's Brain. These findings can be used to support the development of a real-time eye-gaze analysis system to provide personalized feedback and adaptive instructions.

The rest of the paper is organized as follows. Section 2 describes the learning environment. Section 3 describes the proposed methodology to extract content based eye-gaze features from learning environment with multiple hypertext resources. Section 4 describes the experimental design, data collection, methodology to preprocess the data and train the classifiers to predict learning based on features extracted solely from eye-gaze data. The results are reports in section 5. Conclusions, limitations and future work are discussed in section 6.

2. BACKGROUND: THE BETTY'S BRAIN LEARNING ENVIRONMENT

The Betty's Brain learning environment [24] assigns learners the task of teaching a science topic to a teachable agent named Betty by constructing a visual causal map consisting of a set of entities connected by directed causal links. As students build their map, they can ask Betty questions, and can answer them and explain her answers. The students' goal is to teach Betty a causal map that matches a hidden expert model of the topic.

Students' activities are categorized into three primary action types: (1) reading hypertext resources on the science topic (*READ*), (2) building the causal map (*BUILD*), and (3) assessing (*ASSESS*) the correctness of the map [8]. Students iterate among these activities until they have taught Betty a correct model. In this paper, we study learners' information acquisition processes primarily as reading the hypertext resources that describe the science topic under study (*e.g., human causes and effects of climate change*) by breaking it down into a set of subtopics. Each sub-topic describes a system or a process (*e.g., the greenhouse effect*) in terms of entities (*e.g., absorbed heat energy*) and causal relations among these entities (*absorbed heat energy increases the average global temperature*). As students read about the topic, they extract the causal relations between entities and construct the causal map to teach Betty. Figures 1 illustrates the Betty's Brain *READ* (set of hypertext resources) and *BUILD* interfaces.


Betty's Brain - The Teachable Agents Group @ Vanderbilt University
Causal Map
Science Book
Notes
Quiz Results
Teacher's Guide
Betty
Start Conversation
Mr. Davis
Start Conversation
Add a note
Introduction to Climate Change
Introduction to Climate Change
Greenhouse Effect
Solar Energy and Absorbed Lig
Absorbed Light Energy and Abs
Heat Reflected to the Earth
Human Activity
An Earth in Trouble
Deforestation and Carbon Diox
Vehicle Use, Factories, and Ele
One Person's Trash is Another
Impact on Climate
Rising Ocean Levels
Droughts and the Water Cycle
Dictionary
Absorbed Heat Energy
Absorbed Light Energy
Carbon Dioxide
Carrying Capacity
Coastal Flooding
Condensation
Deforestation
Drought
The Greenhouse Effect
Some of the sun's energy is reflected back into space
Greenhouse gases in the atmosphere trap some of the heat
Solar energy passes through the atmosphere warming the Earth

Solar Energy and Absorbed Light

You might have heard about the greenhouse effect on television or in your science classes. That is because the greenhouse effect is important to how the world normally works. By trapping heat generated by the sun, the greenhouse effect keeps the earth warm enough to support life.

The whole process starts with solar energy, which is generated by the sun about 150 million kilometers from the Earth. Most of the sun is made of hydrogen gas. Through a process called **nuclear fusion**, some of the hydrogen in the sun is turned into helium. This releases a huge amount of energy! The energy from nuclear fusion in the sun is radiated into space as electromagnetic waves. **Solar energy** is electromagnetic radiation that carries the energy of the sun to the Earth.

When solar energy reaches the Earth's surface, the Earth absorbs it. Once it has been absorbed, we call it absorbed light energy. The more solar energy that the Earth receives, the more light energy it will absorb. The Earth does not absorb all of the solar energy it receives. Some solar energy is reflected back into space by the atmosphere before ever reaching the Earth's surface. Other solar energy is reflected back into space by sea ice, which is a good reflector of light. When there is more sea ice covering the earth's surface, there will be less absorbed light energy.

(a)

(b)

Figure 1. Betty's Brain system showing (a) *READ* (Science resources) and (b) *BUILD* (Causal Map) Interfaces

Students can assess their own understanding and success in teaching Betty by:

1. Querying Betty using a template for asking *cause-effect* questions. A second pedagogical "mentor" agent, *Mr. Davis*, helps grade Betty's answers by comparing them against the expert model.
2. Asking Betty to take a quiz, which helps them evaluate the current state of the map.

In addition to the three major actions (*READ*, *BUILD*, and *ASSESS*), students can also take *NOTES* on information from the science book, and *CONVERSE* with Betty or Mr. Davis. Students' interactions with the environment are recorded, in log files with associated timestamps.

Student performance in the Betty's Brain environment is measured by their current "map score", which is computed as the difference between the number of correct and incorrect links present in the student's map at any point of time. Depending on the edit actions performed by the student, map score can increase, decrease, or remain the same. Map score patterns vary among students and display their individual learning behaviors.

Students' learning behaviors in Betty's Brain are modeled according to a cognitive/metacognitive task model [19]. Their interactions with the system are mapped to particular skills (for example, reading hypertext resources is mapped to an information acquisition skill), which are then interpreted in terms of the overall learning objectives. A sequential combination of skills, performed in a context, is interpreted as a problem solving strategy. Researchers have employed a combination of analytics methods [34] and exploratory sequence mining techniques for detecting and characterizing students' metacognitive processes [20] in the Betty's Brain environment. Betty's Brain has been shown to significantly improve student learning, as measured by gains observed from pre- to post-tests. [9, 19, 20, 24, 34].

An important component that governs students' learning and causal reasoning processes in Betty's Brain is their ability to interpret the information provided in the hypertext resources and convert it into efficient causal links. However, this information extraction and interpretation procedure cannot be captured completely from our log files. The use of eye tracking devices can help us track the reading behaviors of students and provide more insight into this procedure. Hence, our goal in this work is to use eye tracking devices in classrooms to better understand students' learning behaviors as they interact with Betty's Brain in authentic settings. In the next section, we describe our proposed methodology to extract eye-gaze features that are directly related to content in each of the hypertext resources.

3. METHODOLOGY TO EXTRACT EYE-GAZE FEATURES

The steps involved in extracting content based eye-gaze features from hypertext resources in an open-ended learning environment are shown in Figure 2. In order to extract features, we first align the log data (in Figure 2(a)) from the learning environment and raw data (b) from the eye-tracking device. Then the Area of Interest (AOI) from each section of the hypertext resources (key file) are aligned, and used to extract the content based eye-gaze features. The details of log data and the key file are described below.

Students' interactions with the learning environment are stored with timestamps, in log files. This includes all student activities such as Read, Build, Notes, and Assess actions. To extract the content based eye-gaze features, we define the bounding box coordinates [x, y] of three AOI regions: a) the title, b) the image c) the sentence that explains the causal relationship between entities. The AOI positions vary for each resource page, hence a key file is created with start and end positions of AOI region of each hypertext resource in the learning environment. Table 1 shows a sample key file with details of AOIs for a science resource page *"Solar Energy and Absorbed Light"* [33]. The sentence *"The more solar energy that the Earth receives, the more light energy it will absorb."* describes the causal relationship between the two entities *"Solar energy" and "Absorbed light energy"* that is relevant for the causal model. The [x, y] coordinates of starting position and ending position of the AOIs are identified, for a display with screen resolution of 1600*900, and recorded in the key file.

The raw data from the eye-tracking device contains eye-gaze position on the display represented as [x, y] coordinates with the timestamp for each sample. The number of samples per second are based on the sampling rate of the eye-tracking device. The timestamp in the log data and raw data from eye-tracking are used to align and combine them for further analyses. Using the aligned data and position of AOIs from the key file, the eye-gaze information on AOIs is extracted and then used to extract content based eye-gaze features. Eye movements while reading are measured by fixations (duration of gaze focused on the same point) and saccades (movement of gaze between two fixations) [27, 28]. In this study, we used four frequently used [15, 29] measures of fixation, and two frequently used measures based on saccades as the features as summarized in Table 2. The features are computed for each of the three AOIs discussed above and also for the total page, thus providing a minimum of 4 x 4 = 16 content-based eye-gaze features for each hypertext resource page. Some of the hypertext resources contained multiple sentences that explain the causal relationship between entities.

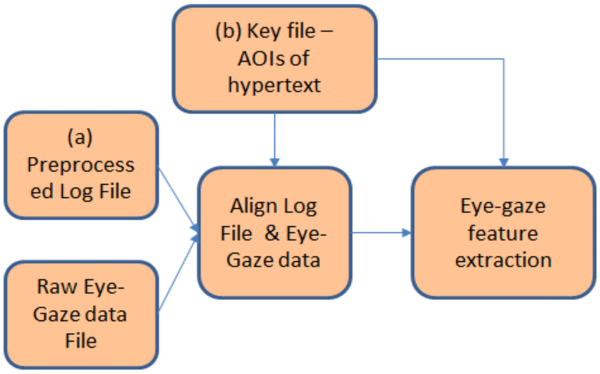

(b) Key file – AOIs of hypertext
(a) Preprocessed Log File
Raw Eye-Gaze data File
Align Log File & Eye-Gaze data
Eye-gaze feature extraction

Figure 2: Algorithm to Extract Content Based Eye-Gaze Features from Multiple Hypertext Resources

Table 1: Sample Key file with AOIs for a resource page

AOI	**Starting position in [x, y] coordinates**	**Ending position in [x, y] coordinates**
Image	[415,350]	[810,640]
Title	[417,120]	[734, 145]
Causal Relation	[416, 281]	[1330,305]
Entities	*Solar energy*	*Absorbed light energy*
Causal Relationship between entities	*The more solar energy that the Earth receives, the more light energy it will absorb.*	

4. EXPERIMENTAL METHODOLOGY

The analysis presented in this paper is based on a recent study of Betty's Brain. The data was collected from eighteen 6th grade students from two classrooms of a middle school in Nashville, Tennessee, USA.

Students used the Betty's Brain system to learn about the causes and effects of climate change. The students' goal was to develop a causal map containing 22 concepts and 25 links representing the greenhouse effect *(e.g. solar energy, absorbed light energy)*, human activities affecting global climate change *(e.g. deforestation,* *vehicle use)*, and impacts on climate *(e.g. sea ice, ocean level, drought).* The hypertext resources were organized into one introductory page, three pages covering the greenhouse effect, four pages covering human activities, and two pages covering impacts on climate. Additionally, a glossary section provided a description of some of the concepts, one per page. The complete resources were made up of 31 hypertext pages.[1]

Table 2: Description of eye-gaze features

Feature	Description
Fixation Count	Total number of fixations counted in a page
Average Fixation Duration in milliseconds	Mean of fixation duration on a page (i.e., Gaze duration mean)
Fixations Count on AOI	Total number of fixations counted in an AOI
Average Fixation Duration on AOI	Mean of fixation duration on AOI
Relative Saccade angle in degrees	The relative angle between two consecutive saccades.
Saccade Amplitude	The size of the saccade measured in degrees or mins of arc

4.1 Study Procedure

The study was conducted over seven school days, with students participating in the study for one 60-minute class period each day. On day 1, students completed the pretest. On day 2, students worked with Betty's Brain introduction topic to get hands-on training on how to identify causal relation with reading text passages. During the second day, we also trained the students on how to calibrate the eye tracker and helped them to create their eye-tracking profile on the laptop. In this study, we used nine Tobii 4c eye-tracking device to collect students' eye-gaze data. The eye trackers were attached to the laptop computer just below the screen using magnetic strips. Students calibrated using the inbuilt Tobii Eye Tracking software[2] that displays on-screen instructions followed by a six point calibration sequence, where the points appear on the screen and disappear when students fixated on each point. Students worked on Betty's Brain climate change topic for four class periods (day 3-6). During these periods, students first selected their eye-tracking profile and calibrated their gaze points using nine-point calibration without the help of researchers. On the last day, students completed the post-test that was identical to the pre-test.

4.2 Data Collection

To extract content based eye-gaze features we combined data from the Tobii 4c eye-tracking devices with log data from Betty's Brain system as they worked on the Climate change topic on days 3-6 of the study.

4.3 Validation of Eye-Tracking Data

Researchers helped the students to set up and calibrate the eye-tracking device during the training day (second session) for a total of 18 students. However, we are not able to use the data from two students' due to continuous calibration failure; hence we used the eye-gaze data collected from 16 students' in this analysis.

On an average, eye gaze data were obtained for 53.3% of the entire duration that each student interacted with the learning environment. The reason for the loss of data can be attributed to students' a) focus on the keyboard while taking notes and typing labels for keywords, b) interaction with other students and c) focus on the teacher or researcher during instructions. To assess the degree to which the proportion of data collected was caused by stable individual differences between students; we correlated the average proportion of data collected over days 1 & 3 for each student with the average duration of data collected over days 2 & 4. This correlation was very strong ($r = 0.89$), demonstrating that factors causing variation in the amount of data collected for each student were strongly affected by individual differences between students. However, given the noisy classroom environment, the overall amount of eye-gaze data collected for 16 of the 18 students was a promising sign that consumer-level eye trackers could be useful in this setting.

4.4 Data Analysis and Methodology

We processed the eye-gaze data using pygaze analyzer, an open-source toolbox for eye-tracking [8] to extract fixation and saccades. The key file, as shown in table 1, is developed based on AOIs in ten hypertext resources. Eye-gaze features as described in table 2 are extracted using the data collected from 16 students.

To predict learners' performance in the map building activity using only the eye gaze data, we considered the map-building activities (*ADD, EDIT and DELETE* causal links) that were immediately followed by a supported[3] by hypertext Read actions [34]. The research methodology to model learners' performance using eye-gaze data on hypertext resources during Read action is shown in Figure 3. The eye-gaze features extracted during each Read action, and performance on the subsequent supported Build actions were used as a labeled data to train and validate the classifier. The trained classifier was then applied on eye-gaze features extracted during all Read actions to classify the learner's reading behavior on hypertext resources as effective or ineffective. The average number of effective and ineffective Read actions over a session were then used to model learners' performance on causal map building actions in the same session.

5. RESULTS

In this section, we first describe the results of eye-gaze feature extraction and performance of the classifier trained using labeled data. Then the analysis of modeling learners' performance using reading behavior is discussed.

[1] The Betty's Brain system can be downloaded from https://wp0.vanderbilt.edu/oele/software/

[2] The Tobii Eye Tracking software was downloaded from https://tobiigaming.com/getstarted/

[3] The two sequential actions Read → Build, is considered supported, only if the information acquired in Read action is used in the Build action.

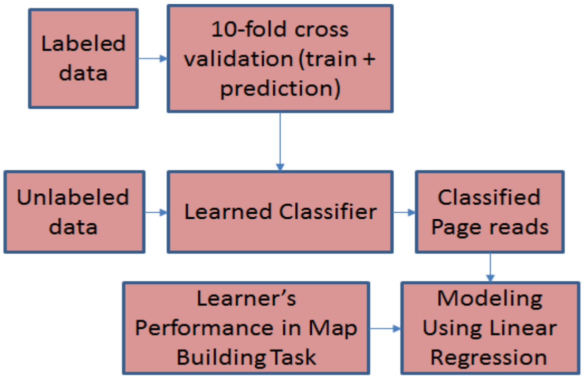

Labeled data
10-fold cross validation (train + prediction)
Unlabeled data
Learned Classifier
Classified Page reads
Learner's Performance in Map Building Task
Modeling Using Linear Regression

Figure 3: Research Methodology to Predict Learning from learner's Reading Behavior

We extracted eye-gaze features during 160 Read actions that were immediately followed and supported by Build actions from 16 student's log and eye-tracking data. Out of 160 eye-gaze features, 36 (22%) were removed due to insufficient eye-gaze data (total duration of eye-gaze on page < 1 millisecond). Of the remaining 124 eye-gaze features collected during Read actions, 104 Build actions were correct, resulting in an increased map score, and only 20 edit actions resulted in a decrease in performance. In order to develop a classifier model using this imbalanced dataset we used Synthetic Minority Over-sampling Technique (SMOTE) algorithm [6], to up-sample the minority data (incorrect edits). SMOTE is used to avoid overfitting when replicating the minor samples during up-sampling. In SMOTE, a subset of data is taken from the minority class to create a synthetic similar instances which are then added to the original dataset.

We used the Gradient tree boosting algorithm [11] for predicting map edit action. In this algorithm, many classification models are trained sequentially, and the loss function of each model is minimized using a gradient descent method. In this analysis, we used decision trees as the classification model for gradient boosting. We used Rapidminer [13] for implementing upsampling and Gradient tree boosting. The classification results using 10 fold cross-validation are shown in Table 3.

The gradient tree boosting algorithm predicted the correctness of map edit action with an accuracy of 80.83%, Cohen's kappa κ = 0.62, and F1 Score = 0.82.

Table 3: Predicting Performance on Map Edit Actions.

	Actual		
Predicted	**Map Edit (+)**	**Map Edit (-)**	**Class Precision**
Map Edit (+)	79	15	84.04%
Map Edit (-)	25	89	78.07%
Class Recall	75.96%	85.58%	

The trained gradient tree boosting classifier was then used to classify learners' reading behavior as effective or ineffective using eye-gaze data during from all of the Read actions. We extracted 1987 eye-gaze features during Read actions of all students. Out of 1987 Reading behaviors extracted, 329 (16.5%) were classified as ineffective and rest were classified as effective. Without applying any up sampling technique, for each student, we computed the number of effective and ineffective read actions per session. To model learners' performance in map building actions using their reading behavior on hypertext resources, we used a linear regression with the net change in map scores per session as a dependent variable. The regression statistics are described in Table 4.

Table 4: Regression Statistics

Multiple R	0.515
R Square	0.262
Adjusted R Square	0.229
Standard Error	3.675
Observations	49

Learner's performance in the map building task could be predicted from a number of effective and effective Read actions by using the following formula:

*Performance = 0.17 * # of effective page Read actions + 0.21 * # of Ineffective Page Read actions - 1.46;   R = 0.51.*

The correlation value, *R*, indicates a moderate degree of correlation between the independent variable (Number of effective and ineffective read actions) and dependent value (Performance in the map building actions).

The results of classifier models trained using the imbalanced data show that prediction of learners' performance for each link-creation event, only using content-based eye-gaze features, was significantly greater than chance (Kappa score κ = 0.62, and F1 Score = 0.82). The results of the linear regression model indicate the ability to predict learner's performance on map building tasks based on their reading behaviors observed during Read actions.

6. CONCLUSIONS, LIMITATIONS AND FUTURE WORK

Our goals in this research were threefold: (1) to test the effectiveness of using consumer-level eye-tracking devices in a noisy classroom environment; (2) to extract the content level eye-gaze features during learners reading hypertext resources in the learning environment; and (3) to predict the learner's performance based on their reading behavior. In this study, we collected eye-gaze data from 16 middle school student while working on Betty's Brain learning environment in a noisy classroom environment. We proposed a methodology to extract content level eye-gaze features and applied it to the data collected from our study. The extracted features were able to predict learner's performance in map building task with an F1 score of 0.82. These results show the ability to track and predict learner's performance that can be used to provide real-time feedback and adaptive instructions to them.

The present study has two limitations. First, we were able to extract only 124 eye-gaze features during the reading task to train the classifier to predict learning. Also, the eye-gaze features extracted were imbalanced necessitating use of an upsampling technique to train and validate the classifier. Second, we were able to collect eye-tracking data only for 54% of the entire duration that student's interaction with the learning environment in the real classroom setting due to the unstructured nature of the environment.

In addition to collecting more data in our future studies, we propose to analyze students' learning behaviors not only from their reading behaviors, but also from learner's other interactions with the system, such as analyzing the quiz answers and interactions with the two virtual agents in the system -- the Mentor, Mr. Davis, and the Teachable Agent, Betty. The goal is to derive more precise information of the coherence relations between actions (see [34]). We also propose to implement real-time eye-gaze analysis to provide personalized feedback based on learner's reading behavior.

7. ACKNOWLEDGMENTS

This project was supported by NSF grant #1623625 to Dan T. Levin (PI) and Gautam Biswas (co-PI).

8. REFERENCES

[1] Amadieu, F., Van Gog, T., Paas, F., Tricot, A. and Mariné, C., 2009. Effects of prior knowledge and concept-map structure on disorientation, cognitive load, and learning. *Learning and Instruction*, *19*(5), pp.376-386.

https://doi.org/10.1016/j.learninstruc.2009.02.005

[2] Basu, S. and Biswas, G., 2016. Providing adaptive scaffolds and measuring their effectiveness in open ended learning environments. *In Proceedings of International Society of the Learning Sciences.*

[3] Biswas, G., Segedy, J.R. and Bunchongchit, K., 2016. From design to implementation to practice a learning by teaching system: Betty's Brain. *International Journal of Artificial Intelligence in Education*, *26*(1), pp.350-364.

[4] Bondareva, D., Conati, C., Feyzi-Behnagh, R., Harley, J.M., Azevedo, R. and Bouchet, F., 2013, July. Inferring learning from gaze data during interaction with an environment to support self-regulated learning. In *International Conference on Artificial Intelligence in Education* (pp. 229-238). Springer, Berlin, Heidelberg

[5] Bransford, J.D. and Schwartz, D.L., 1999. Chapter 3: Rethinking transfer: A simple proposal with multiple implications. *Review of research in education*, *24*(1), pp.61-100.

[6] Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, pp.321-357.

[7] Conati, C., Aleven, V. and Mitrovic, A., 2013. Eye-Tracking for Student Modelling in Intelligent Tutoring Systems. Design recommendations for intelligent tutoring systems, 1, pp.227-236. Vancouver.

[8] Dalmaijer, E.S., Mathôt, S., & Van der Stigchel, S. (2013). PyGaze: an open-source, cross-platform toolbox for minimal-effort programming of eye tracking experiments. Behaviour Research Methods. doi:10.3758/s13428-013-0422-2

[9] Davis, J., Leelawong, K., Belynne, K., Bodenheimer, B., Biswas, G., Vye, N. and Bransford, J., 2003, January. Intelligent user interface design for teachable agent systems. In *Proceedings of the 8th international conference on Intelligent user interfaces* (pp. 26-33). ACM. DOI= 10.1145/604045.604054

[10] D'Mello, S., Kopp, K., Bixler, R.E. and Bosch, N., 2016, May. Attending to attention: Detecting and combating mind wandering during computerized reading. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (pp. 1661-1669). ACM. DOI= 10.1145/2851581.2892329

[11] Friedman, J.H., 2002. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, *38*(4), pp.367-378.

[12] George E. Raptis, Christina Katsini, Marios Belk, Christos Fidas, George Samaras, and Nikolaos Avouris. 2017. Using Eye Gaze Data and Visual Activities to Infer Human Cognitive Styles: Method and Feasibility Studies. *In Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization(UMAP '17).* ACM, New York, NY, USA, 164-173. DOI= 10.1145/3079628.3079690

[13] Hofmann, M. and Klinkenberg, R. eds., 2013. *RapidMiner: Data mining use cases and business analytics applications*. CRC Press.

[14] Hutt, S., Mills, C., Bosch, N., Krasich, K., Brockmole, J. and D'Mello, S., 2017, July. Out of the Fr-Eye-ing Pan: Towards Gaze-Based Models of Attention during Learning with Technology in the Classroom. *In Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization* (pp. 94-103). ACM. DOI= 10.1145/3079628.3079669

[15] Jacob, R.J. and Karn, K.S., 2003. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. In *The mind's eye* (pp. 573-605).

[16] Johnson, C.I. and Mayer, R.E., 2012. An eye movement analysis of the spatial contiguity effect in multimedia learning. *Journal of Experimental Psychology: Applied*, *18*(2), p.178.

[17] Jonassen, D. and Land, S. eds., 2012. Student centered learning environments: foundations, assumptions and design. *Theoretical foundations of learning environments*, pages 3–25.

[18] Kardan, S. and Conati, C., 2013, June. Comparing and combining eye gaze and interface actions for determining user learning with an interactive simulation. In *International Conference on User Modeling, Adaptation, and Personalization* (pp. 215-227). Springer, Berlin, Heidelberg.

[19] Kinnebrew, J.S., Segedy, J.R. and Biswas, G., 2014. Analyzing the temporal evolution of students' behaviors in open-ended learning environments. *Metacognition and learning*, *9*(2), pp.187-215.

[20] Kinnebrew, J.S., Segedy, J.R. and Biswas, G., 2017. Integrating model-driven and data-driven techniques for analyzing learning behaviors in open-ended learning environments. *IEEE Transactions on Learning Technologies*, *10*(2), pp.140-153. DOI= 10.1109/TLT.2015.2513387

[21] Lai, M.L., Tsai, M.J., Yang, F.Y., Hsu, C.Y., Liu, T.C., Lee, S.W.Y., Lee, M.H., Chiou, G.L., Liang, J.C. and Tsai, C.C., 2013. A review of using eye-tracking technology in exploring learning from 2000 to 2012. *Educational research review*, *10*, pp.90-115.

DOI = http://dx.doi.org/10.1016/j.edurev.2013.10.001

[22] Lallé, S., Taub, M., Mudrick, N.V., Conati, C. and Azevedo, R., 2017, June. The Impact of Student Individual Differences and Visual Attention to Pedagogical Agents During Learning with MetaTutor. In *International Conference on Artificial Intelligence in Education* (pp. 149-161). Springer, Cham.

[23] Lee, F.J. and Anderson, J.R., 2001. Does learning a complex task have to be complex?: A study in learning decomposition. *Cognitive psychology*, *42*(3), pp.267-316. DOI= https://doi.org/10.1006/cogp.2000.0747

[24] Leelawong, K. and Biswas, G., 2008. Designing learning by teaching agents: The Betty's Brain system. *International Journal of Artificial Intelligence in Education*, *18*(3), pp.181-208.

[25] Peter A Frensch and Joachim Funke., 2014 Complex problem solving: The European perspective. Psychology Press.

[26] Peterson, J., Pardos, Z., Rau, M., Swigart, A., Gerber, C. and McKinsey, J., 2015, June. Understanding student success in chemistry using gaze tracking and pupillometry. In International Conference on Artificial Intelligence in Education (pp. 358-366). Springer, Cham.

[27] Rayner, K., 1998. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, *124*(3), p.372.

[28] Rayner, K., 2009. The 35th Sir Frederick Bartlett Lecture: Eye movements and attention in reading, scene perception, and visual search. *Quarterly journal of experimental psychology*, *62*(8), pp.1457-1506.

[29] Scheiter, K., Schubert, C. and Schüler, A., 2018. Self- regulated learning from illustrated text: Eye movement modelling to support use and regulation of cognitive processes during learning from multimedia. *British Journal of Educational Psychology*, *88*(1), pp.80-94. DOI=10.1111/bjep.12175

[30] Schoenfeld, A.H., 2010. *How we think: A theory of goal-oriented decision making and its educational applications*. Routledge.

[31] Schwartz, D.L., and Arena, D., 2013. *Measuring what matters most: Choice-based assessments for the digital age*. MIT Press.

[32] Sengupta, P., Kinnebrew, J.S., Basu, S., Biswas, G. and Clark, D., 2013. Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, *18*(2), pp.351-380.

[33] Segedy, J.R., 2014. *Adaptive scaffolds in open-ended computer-based learning environments*. Vanderbilt University.

[34] Segedy, J.R., Kinnebrew, J.S. and Biswas, G., 2015, June. Coherence over time: understanding day-to-day changes in students' open-ended problem solving behaviors. In *International Conference on Artificial Intelligence in Education*(pp. 449-458). Springer, Cham.

[35] Steichen, B., Wu, M.M., Toker, D., Conati, C. and Carenini, G., 2014, July. Te, Te, Hi, Hi: Eye gaze sequence analysis for informing user-adaptive information visualizations. In *International Conference on User Modeling, Adaptation, and Personalization* (pp. 183-194). Springer, Cham.

Does Deep Knowledge Tracing Model Interactions Among Skills?

Shirly Montero*
Dept of Computer Science
University of Colorado Boulder
shmo8450@colorado.edu

Akshit Arora*
Dept of Computer Science
University of Colorado Boulder
akshit.arora@colorado.edu

Sean Kelly
Woot Math
Boulder, Colorado
sean.kelly@wootmath.com

Brent Milne
Woot Math
Boulder, Colorado
brent.milne@wootmath.com

Michael Mozer
Dept of Computer Science
University of Colorado Boulder
mozer@colorado.edu

ABSTRACT

Personalized learning environments requiring the elicitation of a student's knowledge state have inspired researchers to propose distinct models to understand that knowledge state. Recently, the spotlight has shone on comparisons between traditional, interpretable models such as Bayesian Knowledge Tracing (BKT) and complex, opaque neural network models such as Deep Knowledge Tracing (DKT). Although DKT appears to be a powerful predictive model, little effort has been expended to dissect the source of its strength. We begin with the observation that DKT differs from BKT along three dimensions: (1) DKT is a neural network with many free parameters, whereas BKT is a probabilistic model with few free parameters; (2) a single instance of DKT is used to model all skills in a domain, whereas a separate instance of BKT is constructed for each skill; and (3) the input to DKT interlaces practice from multiple skills, whereas the input to BKT is separated by skill. We tease apart these three dimensions by constructing versions of DKT which are trained on single skills and which are trained on sequences separated by skill. Exploration of three data sets reveals that dimensions (1) and (3) are critical; dimension (2) is not. Our investigation gives us insight into the structural regularities in the data that DKT is able to exploit but that BKT cannot.

Keywords

Personalized learning, Online education, Knowledge tracing, Deep learning, Sequential modeling

1. INTRODUCTION

The optimization of the human learning is a recurring topic in educational research. Traditional human instructors monitor and assess a student's knowledge and adapt instructional activities to help the student achieve her goals. Assuming the knowledge in a domain has been decomposed in a hierarchy of skills, the sequence of learning activities becomes a scaffold for the learning process, helping the student to acquire prerequisite skills before moving to more complex skills in the hierarchy [1]. Therefore, in tailoring the sequence to the needs of the student it is essential to track, assess, and predict the student's changing knowledge state, thereby personalizing the design. In reality, with limited educational resources a standardized lesson design is more the norm than the exception. Nevertheless, automated tutoring/self-study designs have presented an interesting attempt to personalize learning, and offer a more budget-friendly option in the long term. To be effective, automated tutoring systems should model the student's knowledge state, known as *knowledge tracing* [3], substituting the cues that a human instructor would use to assess the student with the student's performance along the sequence of formative and summative learning activities. However, knowledge tracing and the evaluation of the personalized learning environments remain a complex endeavor and the focus of interest for applied machine learning research.

1.1 Knowledge Tracing

A knowledge-tracing model tracks a student's evolving knowledge state as the student practices a sequence of problems [3]. The knowledge state is decomposed into a set of domain *skills* required to solve the specific problems that the student is attempting. Each problem is labeled with the corresponding skill required for that problem. The critical data to be modeled thus consist of a sequence of pairs, $\mathcal{D}_q = \{\ldots(X_{qt}, Y_{qt})\ldots\}$, where X_{qt} is a categorical random variable indicating the specific skill required to be able to solve the problem presented to student q on trial t, and Y_{qt} is a binary random variable denoting the outcome of the trial, with $Y_{qt} \in \{\mathit{correct}, \mathit{error}\}$. Of course, most modern data sets have far richer information—the use of supporting materials or hints, response latencies, time between trials, number of attempts, the specific problems being attempted, etc. For the present research, we are not considering these additional sources of data.

*Denotes equal contribution by authors

In Bayesian knowledge tracing (BKT), the data are partitioned by skill, leading to a skill specific dataset,

$$\mathcal{D}_{qs} = \{(X_{qt}, Y_{qt})|X_{qt} = s\}$$

in which the trial sequence is re-indexed for each skill s. BKT is a hidden Markov model that performs inference to determine a latent binary skill variable K_{qst}, denoting the knowledge state of student q on skill s at the start of trial t. The model for skill s has 4 parameters [5], $\boldsymbol{\theta}_s$, with the following interpretations in terms of the model:

$$\begin{aligned}\boldsymbol{\theta}_s =&\{P(K_{qs0}) = 1, \quad P(Y_{qt} = 1|K_{qst} = 0),\\ &1 - P(Y_{qt} = 0|Kqst = 1), P(K_{qst} = 1|K_{q,s,t-1} = 0\}.\end{aligned}$$

In this form the model assumes no forgetting, i.e., the knowledge state K cannot transition from 1 to 0. Note that each skill is treated independently; cross-skill interactions are not modeled.

DKT [5, 7] is a recurrent neural network whose input layer is a representation of the previous trial, $(X_{q,t-1}, Y_{q,t-1})$ and whose output layer is a prediction, for every possible skill, of whether the student would answer problems of that skill correctly, i.e., $\forall s$, $P(Y_{q,t}|X_{q,t} = s)$.[1] Internally, DKT has a layer of recurrent hidden units that, through training, learn to hold the student's knowledge state in order to make predictions. Typically, the hidden layer contains LSTM units, often used to handle sequence processing tasks because of their ability to maintain state over time.

As originally implemented, DKT makes three assumptions that distinguish it from BKT:

1. All skills are interleaved in a single sequence over time, and predictions are made for each trial in the sequence. In contrast, BKT assumes that skills are presented in separate sequences. We will refer to this distinction as *combined sequence* (*CS*) versus *separate sequences* (*SS*).
2. All skills are learned by a single model that combines information across skills. In contrast, BKT assumes that a separate model is trained on each skill, and thus the parameters for different skills do not interact. We refer to this distinction as *combined model* (*CM*) versus *separate models* (*SM*).
3. DKT is of course based on a neural network, whereas BKT is a probabilistic model. The neural network has far greater flexibility. For example, BKT assumes that once a student learns they stay in the 'knowing' state. In contrast, DKT can model forgetting. To illustrate another difference, DKT can in principle remember the last n trials and condition its prediction on this complex state representation, whereas BKT is Markovian—it embodies the input history in a single binary state variable.

Assumption 1 is conditioned on assumptions 2 and 3; assumption 2 is conditioned on assumption 3. Our goal is to tease apart these assumptions and examine them individually, allowing us to determine which assumptions are most responsible for the improvements in performance that DKT achieves over BKT. In addition to the standard form of BKT and DKT, we introduce two new variants of DKT: one that drops assumption 1, and one that drops assumptions 1 and 2. For the sake of understanding the relationship among the four models, we relabel the standard forms of BKT and DKT, obtaining the following progression of models:

- **DKT-CM-CS**: The standard form of DKT, which is a single neural network that learns all skills (the combined model or CM) and its input sequence consists of the interlaced sequence of trials across all skills (the combined sequence or CS). This model incorporates assumptions 1-3.
- **DKT-CM-SS**: DKT minus assumption 1. This variant is trained on a separate sequence for each skill. A single model is still used to predict for all skills (the combined model or CM) but the input is separated by skill (the separate sequences or SS).
- **DKT-SM-SS**: DKT minus assumptions 1 and 2. This variant trains a different model for each skill (separate models or SM) and because each skill is fed into a different model, it is necessary to separate the sequences by skill (SS).
- **BKT-SM-SS**: The standard form of BKT. We augment the name with SM to remind the reader that a separate instantiation of the model is constructed for each skill, and with SS to indicate that sequences are separated by skill and fed into the corresponding model. This model drops all three assumptions of DKT.

Pairwise comparisons among models allow us to examine individual assumptions: DKT-CM-CS and DKT-CM-SS differ only in assumption 1; DKT-CM-SS and DKT-SM-SS differ only in assumption 2; and DKT-SM-SS and BKT-SM-SS differ only in assumption 3. By examining the performance differences between each pair, we can determine the value of each assumption.

1.2 Related Work

Recent studies compare traditional models such as Bayesian Knowledge Tracing (BKT) and its variants against complex neural network models such as Deep Knowledge Tracing (DKT) [4, 5, 6, 7, 8, 10, 11, 12]. The basic BKT (or BKT-SM-SS) is at a distinct disadvantage relative to the standard DKT (or DKT-CM-CS) when it comes to exploiting inter-skill similarities, integrating recency effects, contextualizing trials and representing variations on the student's abilities. Therefore, DKT on balance outperforms basic BKT. Efforts have been made to show that when additional machinery is added to BKT, it rises in performance to a comparable level

[1] The inputs and outputs of DKT can be representations of either *skills* or *problems*. For example, DKT could represent 4 + 3 and 7 + 2 as two distinct problems or it could represent them as the skill *single-digit addition*. Because BKT operates with the level of representation being skills and we wish to compare DKT to BKT, our implementation of DKT does the same: its representation of the current trial is a skill index and the correctness of the response; its representation of the output is one prediction per skill index.

with DKT [5]. But little has been done to examine what factors are contributing to the superiority of DKT.

2. METHODOLOGY

2.1 Data sets

We examined three data sets which vary in the number of students and the number of skills. Two data sets, ASSISTments 09-10(b) and KDD Cup 2010, are well studied in the educational data mining literature. The ASSISTments data set is generated from an online grade school mathematics tutor. The 09-10(b) version of the data were cleaned by Xiong et al. [11] to remove repeated multiple skill problems which, in the original data base, were duplicated for each component skill, and when left in the data set give an advantage to DKT over BKT. ASSISTments 09-10(b) consists of 4217 students and 124 skills. The KDD Cup 2010 data is from the 2005-2006 Cognitive Algebra Tutor [9]. These data consist of 574 students and 100 skills. Both data sets were obtained from a GitHub repository of one of the authors [13]. The data in the repository are divided into a training and a test set.

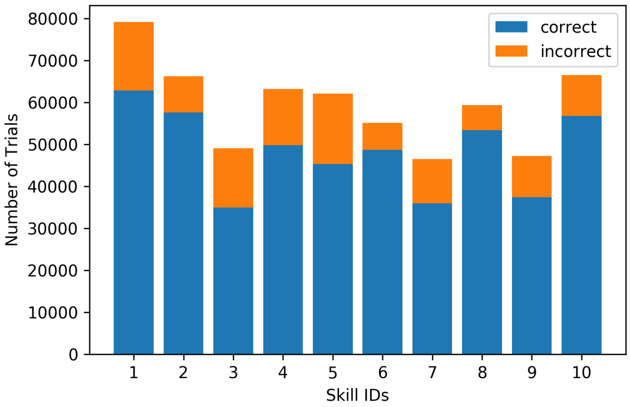


Figure 1: Example distribution of the trials in Woot Math dataset among the skills and the correctness of their outcomes used for training.

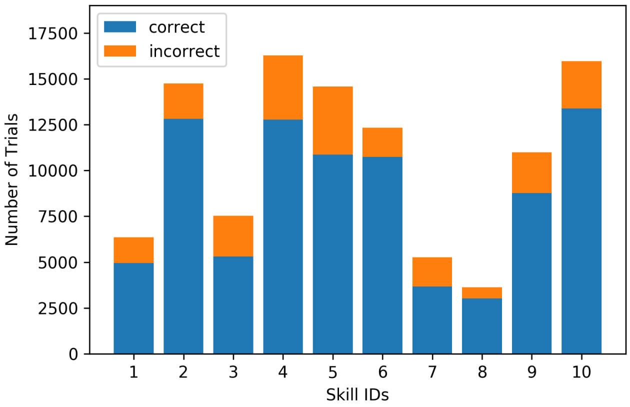


Figure 2: Example distribution of the trials in Woot Math dataset among the skills and the correctness of their outcomes used for evaluation.

The third data set was collected by Woot Math, a Boulder Colorado start up that develops adaptive learning environments for mathematics. The focus is on helping students in grades 3-8 master core math concepts, beginning with rational numbers. The Woot Math software delivers a personalized progression of interleaved video instruction and scaffolded problems to mimic the natural give-and-take between a student and a tutor. The content within the environment is divided in units. Each unit is a collection of lessons related to a specific area of a subject from the elementary mathematics curriculum, e.g, fractions. Further, each lesson comprises several sets of problems and instructional content that focus on a particular aspect of a unit. Ultimately, each problem set is coupled to a skill. It is worth noting that the learning trajectories are adaptive, and as consequence different students have different numbers of trials in a lesson.

That dataset consists of anonymized data capturing the state of the learning platform when the student interacted with a particular labeled exercise. Although more secondary data features are available, we selected only the following as the primary features: an identifier tag, which is an unique identifier for a skill, and the correctness of the answer of binary outcome of the interaction. In order to decrease sparsity, we limited our data set to those students who had at least 50% of their trials within ten most popular skills completed. This selection rendered a set of 625,619 trials from 11,659 students, with exercises drawn from among 10 skills. The data were split by student to obtain an 80:20 ratio of training to testing examples (9,327 students for training, 2,332 students for testing). The distributions of trials among the ten skills and the correctness of their outcomes are shown in Figure 1 for data used during training and Figure 2 for data used during the evaluation.

2.2 Data encoding

For DKT, input vectors are a one-hot encoding of the previous trial, specified by the conjunction of (a) the skill to which the trial belonged, and (b) whether or not the trial was answered correctly. Thus, if there are n_s skills in total, then the input vector has $2n_s$ units, exactly one of which would be turned on for any input. The input vectors are fed into the models in a sequence sorted by the temporal order of the trials. The output from DKT is a vector with n_s elements, each element s being the model's estimate of the probability that the student had acquired skill s given the performance history of the student. This probability also specifies how likely the student will be to answer the next trial correctly if it is a problem requiring skill s.

2.3 Model implementation

To implement DKT methods, we modified the source code used by Xiong et al. [11], as obtained from one of the author's GitHub repository [13]. The modifications pertained the way the data were to be fed to the model when single skill sequences were used. For DKT, we ran five replications of training the neural net with different random weight initializations each replication. The same training and test split was used for all five replications.

For initializing weights in all the DKT methods, we used random uniform weights in the range $[-.05, +.05]$. All DKT models had a single hidden layer. DKT-SM-SS used 10 LSTM units for all datasets. For DKT-CM-SS and DKT-CM-CS, we used one hidden layer with 50 LSTM units for the Woot Math dataset and 200 LSTM units for the other two datasets, due to the fact that they contain more skills. Additionally, for all DKT models, we used drop-out on the hidden layer with keep probability of 0.6.

Rather than run BKT on ASSISTments and KDD, we report the results from Xiong et al. [11]. Our own implementation of BKT was used to obtain performance estimates for the Woot Math data.

3. RESULTS

We estimate the discriminative performance of each model—its ability to predict when a student will answer correctly or incorrectly—using the signal detection AUC (area under the curve) measure. There are a two methods by which AUC can be computed. One method, within-skill AUC, involves separating the test data for all students by skill and computing an AUC value for each skill and then computing the mean across skills. The other method, between-skill AUC, involves combining data from all students and all skills and computing a single AUC score. In general, the between-skill AUC is larger than the within-skill AUC for two reasons. First, it incorporates the degree to which models are successful at predicting relative performance among skills. Second, the between-skill AUC weighs all trials equally, whereas the within-skill AUC de-emphasizes skills with many trials. In our work, we compute between-skill AUC, both because it is sensitive to aspects of the data we care about and it matches the methodology used by Xiong et al. [11].

Table 1 shows a summary of results for the three data sets (rows of the table) and the four models (columns 4-7 of the table). From left to right, BKT-SM-SS is the basic BKT model, for which a separate model is trained per skill and the sequences are separated by skill. DKT-SM-SS is an implementation of DKT in which a separate model is constructed for each skill and the sequences are separated by skill; this procedure is analogous to the manner in which BKT is trained, except the model is a neural network instead. DKT-CM-SS involves a single combined model trained on all skills, but the sequences fed to the model are separated by skill. Finally, DKT-CM-CS is the standard implementation of DKT in which a combined model is trained on all skills and the input sequences combine skills to obtain an interleaved trial history.

3.1 Interleaved- vs. blocked-skill sequences

DKT-CM-CS and DKT-CM-SS differ only in the manner in which the student sequences are parsed. The combined sequences interleave various skills; the separate sequences are blocked or filtered by skill. For example, 1-3-3-2-2-1-1-2-3 is an interleaved sequence, and {1-1-1, 3-3-3, 2-2-2} are the set of blocked sequences. In both cases, the sequence order corresponds to temporal order of the trials. Our results show a win for DKT-CM-CS for ASSISTments and KDD. In these cases, DKT is able to leverage the interaction among skills. One likely form of interaction that the model exploits is the fact that strong students perform well on all skills, weak students perform more poorly on all skills. Consequently, there should be an inter-skill correlation for a given student. To elaborate, consider the sequence of trials with two skills, 1-1-1-2-2-2. If the student performs extraordinarily well on the 1-1-1 sequence, this observation should be predictive of better-than-average performance on 2-2-2. We suspect that adding IRT-like student ability parameters to DKT might eliminate the difference between the combined- and separate-sequence versions of DKT.

For the Woot Math data set, there was no benefit to combining. We hypothesize that the reason for this finding is that there are only 10 skills, and the breakdown by skill is fairly coarse. Because the skills have little in common, there is less likely to be transfer from one skill to another, and therefore predicting performance on one skill would not benefit from knowing performance on another skill. (Similarly, you wouldn't expect, say, someone's driving ability to predict their juggling ability.)

3.2 Combined-skill vs. separated-skill models

Both DKT-CM-SS and DKT-SM-SS are trained on sequences blocked by skill. They differ in that DKT-CM-SS is trained on all skills at once. Thus its parameters are shared across skills. In contrast, a separate instance of DKT-SM-SS is trained for each skill. Thus, its parameters are not shared across skills. In both cases, AUCs are computed by pooling data across skills and computing a single AUC—the between-skill AUC we referred to earlier.

We do not observe a significant difference in performance between DKT-CM-SS and DKT-SM-SS. On KDD they perform almost identically. On ASSISTments, DKT-SM-SS does slightly better. And on Woot Math, DKT-CM-SS does slightly better. In principle, training a combined model on all skills will be beneficial if different skills are learned in a similar fashion, i.e., if the time course of learning skill s_1 is related to the time course of learning skill s_2. When there is similarity across skills, there can be inter-skill transfer in modeling the temporal dynamics of learning. However, the benefit of this transfer should diminish as data sets get larger. With a large enough data set for skill s_1, the weak inductive bias of s_2 provides little benefit. We suspect that the reason for observing no benefit by training a single model on all skills is that our data sets are relatively large. It is possible on much smaller data sets, we would observe a benefit of using data from skill s_1 to constrain predictions on skill s_2.

3.3 Neural network vs probabilistic model

DKT-SM-SS and BKT-SM-SS are trained in exactly the same way: each model has distinct parameters for each skill, and data from one skill is not used to inform performance on other skills. The models differ in that DKT-SM-SS is an intrinsically flexible neural network with hundreds of parameters, whereas BKT-SM-SS has 4 parameters. By restricting our neural network to model only single skills we are taking out of the equation the possibility of exploiting inter-skill similarities, leveling the playing field for the more restricted BKT model. Nonetheless, the results indicate better performance of the neural net than the probabilistic model on all three data sets. This is consistent with the neural net being more flexible in characterizing the time course of learning.

Table 1: Test set performance (AUC) for four models. Standard deviations (N =5) are in parenthesis.

Dataset	# Students	# Skills	BKT-SM-SS	DKT-SM-SS	DKT-CM-SS	DKT-CM-CS
ASSISTments 09-10(b)	4217	124	0.630	0.733 (0.0003)	0.726 (0.0008)	0.809 (0.0021)
KDD	574	100	0.620	0.771 (0.0003)	0.764 (0.0013)	0.818 (0.0025)
Woot Math	11659	10	0.727	0.745 (0.0007)	0.760 (0.0005)	0.745 (0.0032)

BKT-SM-SS embodies a strongly restricted model of learning. For example, BKT-SM-SS assumes that the probability of learning on trial t_1 is identical to the probability of learning on t_2, for any t_1 and t_2. In contrast, DKT-SM-SS might discover that if a student does not learn early on, they are not likely to learn later on.

4. CONCLUSIONS

Our goal in this research is to understand the factors that contribute to the strong performance of DKT. We explored three factors that differentiate DKT and BKT, and we developed a continuum of 4 models which, when paired, allowed us to evaluate one factor at a time. Our three key findings are as follows. First, DKT benefits from being presented with a sequence of interleaved skills. We hypothesize that this benefit is due to being able to estimate strength of a student based on their performance on one skill and then use this estimate to predict performance on another skill. Second, DKT does not benefit per se by learning about multiple skills at once versus learning about a single skill. We speculate that the reason for this finding is that we have relatively large data sets, and the inductive bias provided by one skill offers little leverage in modeling other skills. Third, DKT shows a large benefit by being a flexible model that does not incorporate a strong theory of human learning, as does BKT. This is perhaps our most significant finding, as it suggests that the simple all-or-none learning-without-forgetting theory that BKT posits is too simplistic.

5. ACKNOWLEDGMENTS

The authors wish to thank Dr. Mohammad Khajah for many useful discussions. This research was supported by the National Science Foundation awards EHR-1631428 and SES-1461535.

6. REFERENCES

[1] L. W. Anderson and D. R. Krathwohl, editors. *A Taxonomy for Learning, Teaching, and Assessing. A Revision of Bloom's Taxonomy of Educational Objectives*. Allyn & Bacon, New York, 2 edition, December 2001.

[2] T. Barnes, M. Chi, and M. Feng, editors. *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016, Raleigh, North Carolina, USA, June 29 - July 2, 2016*. International Educational Data Mining Society (IEDMS), 2016.

[3] 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.

[4] Y. Gong, J. E. Beck, and N. T. Heffernan. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *Proceedings of the 10th International Conference on Intelligent Tutoring Systems - Volume Part I*, ITS'10, pages 35–44, Berlin, Heidelberg, 2010. Springer-Verlag.

[5] M. Khajah, R. V. Lindsey, and M. Mozer. How deep is knowledge tracing? In Barnes et al. [2].

[6] A. Lalwani and S. Agrawal. Few hundred parameters outperform few hundred thousand? In X. Hu, T. Barnes, A. Hershkovitz, and L. Paquette, editors, *Proceedings of the 10th International Conference on Educational Data Mining*, EDM '17, pages 448–453, 2017.

[7] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 505–513, 2015.

[8] Y. Qiu, Y. Qi, H. Lu, Z. A. Pardos, and N. T. Heffernan. Does time matter? modeling the effect of time with bayesian knowledge tracing. In M. Pechenizkiy, T. Calders, C. Conati, S. Ventura, C. Romero, and J. C. Stamper, editors, *Proceedings of the 4th International Conference on Educational Data Mining, Eindhoven, The Netherlands, July 6-8, 2011*, pages 139–148. www.educationaldatamining.org, 2011.

[9] J. Stamper, A. Niculescu-Mizil, S. Ritter, G. Gordon, and K. Koedinger. Algebra i 2005-2006. Challenge data set from kdd cup 2010 educational data mining challenge. Find it at http://pslcdatashop.web.cmu.edu/kddcup/downloads.jsp, 2010.

[10] L. Wang, A. Sy, L. Liu, and C. Piech. Deep knowledge tracing on programming exercises. In *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, L@S '17, pages 201–204, New York, NY, USA, 2017. ACM.

[11] X. Xiong, S. Zhao, E. V. Inwegen, and J. Beck. Going deeper with deep knowledge tracing. In Barnes et al. [2], pages 545–550.

[12] Y. Zhang, R. Shah, and M. Chi. Deep learning + student modeling + clustering: a recipe for effective automatic short answer grading. In Barnes et al. [2], pages 562–567.

[13] S. Zhao. 2016-edm. https://github.com/siyuanzhao/2016-edm.

Mining MOOC Lecture Transcripts to Construct Concept Dependency Graphs

Fareedah ALSaad[*]
University of Illinois at
Urbana-Champaign
Urbana-Champaign, USA
alsaad2@illinois.edu

Assma Boughoula
University of Illinois at
Urbana-Champaign
Urbana-Champaign, USA
boughou1@illinois.edu

Chase Geigle
University of Illinois at
Urbana-Champaign
Urbana-Champaign, USA
geigle1@illinois.edu

Hari Sundaram
University of Illinois at
Urbana-Champaign
Urbana-Champaign, USA
hs1@illinois.edu

ChengXiang Zhai
University of Illinois at
Urbana-Champaign
Urbana-Champaign, USA
czhai@illinois.edu

ABSTRACT

This paper addresses the question of identifying a concept dependency graph for a MOOC through unsupervised analysis of lecture transcripts. The problem is important: extracting a concept graph is the first step in helping students with varying preparation to understand course material. The problem is challenging: instructors are unaware of the student preparation diversity and may be unable to identify the right resolution of the concepts, necessitating costly updates; inferring concepts from groups suffers from polysemy; the temporal order of concepts depends on the concepts in question. We propose innovative unsupervised methods to discover a directed concept dependency within and between lectures. Our main technical innovation lies in exploiting the temporal ordering amongst concepts to discover the graph. We propose two measures—the Bridge Ensemble Measure and the Global Direction Measure—to infer the existence and the direction of the dependency relations between concepts. The bridge ensemble measure identifies concept overlap between lectures, determines concept co-occurrence within short windows, and the lecture where concepts occur first. The global direction measure incorporates time directly by analyzing the concept time ordering both globally and within lectures. Experiments over real-world MOOC data show that our method outperforms the baseline in both AUC and precision/recall curves.

Keywords

Concept Dependency Graph, Temporal Order, Bridge Ensemble Measure, Global Direction Measure, Edge Direction, Edge Existence.

1. INTRODUCTION

This paper presents two methods to identify extant concept relationships in lectures from a Massive Open Online Course (MOOC).

The problem of concept relationship discovery within MOCCs will help adapt to learner diversity where students from all over the globe take classes from MOOCs. Developing a fine-grained map of the concepts presented in the MOOC, indicating pre-requisite relationships, can facilitate students browsing into course materials flexibly. In addition, such a map can help in emphasizing the important topics in the course and how they are related, which can help improve students understanding. It can be further used to represent the knowledge state of a student at the concept level, and thus enable personalization in recommending course materials or quiz questions to students. In this paper, our goal is to construct such a map automatically for any course in order to accommodate students' diversity by supporting personalized learning.

Generating such a concept dependency graph presents a number of challenges. First, the instructor cannot predict the prior preparation of the students taking the class or the granularity at which she should develop the concept graph, and ensuring that such a concept graph remains up to date every year is time consuming. Second, an instructor does not introduce concepts in a rigid order, wherein she will always present the prerequisite concept before introducing the main concept; which makes it difficult in determining the presence and the direction of a relationship between concepts.

We propose innovative unsupervised methods to discover a directed concept dependency graph. We use lecture transcripts, as do Chaplot and Koedinger [2], to model the dependency structure between course concepts. Where Chaplot and Koedinger focus on modeling the prerequisite structure between units or lectures, we instead focus on inferring the dependency structure among concepts that appear *within and between* lectures. Our main technical innovation lies in exploiting the temporal ordering amongst concepts to discover the graph. To the best of our knowledge, we are the first to use temporal features to construct the dependency graph. We propose two measures—the Bridge Ensemble Measure and the Global Direction Measure—to infer the existence and the direction of the dependency relations between concepts.Both proposed measures outperform the baseline method [2] in AUC and the precision/recall curves.

[*] King AbdulAziz University, Jeddah, Saudi Arabia.

The rest of the paper is organized as follows. In section 2, we formally frame our problem before describing the two proposed measures in section 3. Section 4 elaborates our approach for the evaluation and section 5 presents some limitations. Finally, we discuss some related work in section 6 before concluding our work on section 7.

2. PROBLEM DEFINITION

Informally, the problem explored in this work can be stated as follows: given course data, predict the dependency relationships between the course concepts. More formally, let X be the course represented by an ordered list of transcripts corresponding to each lecture: $X = [T_1, T_2, \ldots, T_M]$ where M is the total number of lectures. Let C_X be the set of concepts discussed in the course $C_X = \{c_1, c_2, \ldots, c_N\}$, where N is the total number of unique concepts. Given X and C_X, we aim to generate the concept dependency graph that relates concepts in C_X according to their prerequisite relationships. The resulting concept dependency graph is described by an edge weight matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$. Each entry a_{ij} of matrix $\mathbf{A}$ will contain the edge weight for the associated relationship $c_i \rightarrow c_j$, which means concept c_i is a prerequisite for concept c_j. The edge weight reflects the level of confidence in the inferred relationship. Notice that since the prerequisite relationship has a direction, $\mathbf{A}$ is **not** symmetric.

$$\mathbf{A} = \begin{bmatrix} 0 & \ldots & \ldots & W(c_1 \rightarrow c_N) \\ W(c_2 \rightarrow c_1) & 0 & \ldots & W(c_2 \rightarrow c_N) \\ \ldots & \ldots & \ldots & \ldots \\ W(c_N \rightarrow c_1) & W(c_N \rightarrow c_2) & \ldots & 0 \end{bmatrix}$$

The problem of constructing the concept dependency graph can be reduced to the problem of computing the edge weight between pairs of concepts given course data.

3. LINKING COURSE CONCEPTS

To relate the course concepts according to their dependency relationships, we propose two measures: the Bridge Ensemble Measure and the Global Direction Measure.

3.1 Bridge Ensemble Measure

The Bridge Ensemble Measure (BEM) captures concept dependency structure utilizing inter-lecture and intra-lecture strategies. It contains three components: Bridges, Sliding Windows, and the First Lecture Indicator.

3.1.1 Bridges

Let us look at how instructors naturally introduce concepts and their prerequisite(s). Let C_X be the set of concepts presented in course X and let c_a and c_b be concepts in that set. Determining the presence of a concept c_a in a lecture transcript T_i is discussed further in section 4.1. Suppose that c_a is a prerequisite to c_b. Then it stands to reason that (1) c_a will be introduced before c_b in the course progression, and (2) while explaining or talking about c_b, the instructor will naturally refer to c_a.

Bridge concepts allow us to exploit the temporal nature of lectures to infer concept dependency relationships across lectures. Intuitively, bridge concepts are introduced in an earlier lecture but re-appear in a later lecture when some new concept(s) are introduced. Accordingly, bridge concepts signal a prerequisite relationship from the bridge concepts to the new concepts introduced in the later lecture. For example, in Figure 1, the bridge concepts c_3 and c_4 are more likely to be prerequisite to concepts c_5, c_6, and c_7 discussed in lecture L_2. Formally, let L_i be the set of concepts in the lecture i in course X, and L_j be the set of concepts for the lecture j where $j > i$. The intersection $L_j \cap L_i$ contains all the concepts that appear in both lectures. We call these **bridge concepts**. The difference $L_j \setminus L_i$ contains **difference concepts** which are the concepts present in the later lecture j but not in the earlier lecture i. If c_a belongs to the bridge concepts and c_b belongs to the difference concepts, then there is evidence for the dependency relationship $c_a \rightarrow c_b$ and the edge weight $W(c_a \rightarrow c_b)$ should increase. As a result, the bridge set $B_{ji} = \{(c_a \rightarrow c_b) \mid c_a \in L_j \cap L_i \wedge c_b \in L_j \setminus L_i\}$ contains all candidate prerequisite edges from lecture L_i to lecture L_j. If we replicate this exercise for every possible pair of lectures, we will end up with a comprehensive set of all possible candidate bridge edges **Bridges** for the course:

$$\mathbf{Bridges} = B_{M(M-1)} \cup B_{M(M-2)} \cup \ldots \cup B_{21} \qquad (1)$$

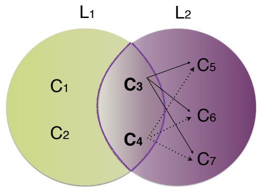


Figure 1: *The bridging concepts (c_3 and c_4) between lecture L_1 and L_2 and the resulting candidate prerequisite relationships.*

To calculate the edge weight of candidate edges in **Bridges**, we use the following bridge scoring function

$$W(c_a \rightarrow c_b) \approx F_{\mathbf{Bridges}}(c_a \rightarrow c_b) \qquad (2)$$

where

$$F_{\mathbf{Bridges}}(c_a \rightarrow c_b) = \frac{\text{The number of lectures where we observe both } c_a \text{ and } c_b}{\text{The number of lectures where we observe } c_b} = \frac{|\{L_j | c_a, c_b \in L_j\}|}{|\{L_j | c_b \in L_j\}|}. \qquad (3)$$

Keep in mind that the bridge scoring function will only calculated for candidate edges belong to **Bridges**. Other pairs of concepts will have zero value for the bridging score.

3.1.2 Sliding Windows

Bridge edges determined by the Bridge Method do not capture every possible prerequisite relationship. Consider the case where concept c_b has a strong prerequisite c_a, but c_a and c_b only appear together either in the set of bridge concepts ($L_j \cap L_i$) or in the set of difference concepts ($L_j \setminus L_i$). As a result, $c_a \rightarrow c_b$ will never appear in **Bridges** and hence the Bridge method cannot infer the prerequisite relations between them.

To solve this problem and capture intra-lecture prerequisite relationships, we zoom into each lecture and consider the proximity of concepts being presented in the lecture. Let $\vec{L_j} = [c_1, c_2, ..., c_n]$ be an ordered list of concepts discussed in lecture j, where n is the total number of concepts. Keep in mind that this ordered list contains redundant concepts which appear in the order where the instructor mentioned them. In the sliding windows method, we segment $\vec{L_j}$

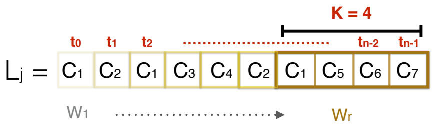


Figure 2: *A visualization example of lecture $\vec{L_j}$ with $r = n - K + 1$ sliding windows of size $K = 4$. The sliding windows captures the proximity of concepts.*

into windows $W_i = [c_i, \ldots, c_{i+K-1}]$ as follows:

$$\textbf{Windows}_j = \begin{cases} \{W_i \mid 1 \le i \le n - K + 1\} & n \ge K \\ \left\{\vec{L_j}\right\} & n < K. \end{cases} \quad (4)$$

Figure 2 depicts the representation of lecture $\vec{L_j}$ using $r = n - K + 1$ windows of size $K = 4$. In this study, we choose the K that gives the best performance; K is set to *10* concepts.

The more windows in which c_a and c_b appear together, the stronger the relationship between c_a and c_b is; thus the edge weight should increase. The second component of the BEM for edge weights is the probability of the edge $c_a \to c_b$ given the information we have about all windows in all lectures $\textbf{Windows} = \bigcup_j \textbf{Windows}_j$.

$$W(c_a \to c_b) \approx F_{\textbf{Bridges}}(c_a \to c_b) + F_{\textbf{Windows}}(c_a \to c_b) \quad (5)$$

Where:

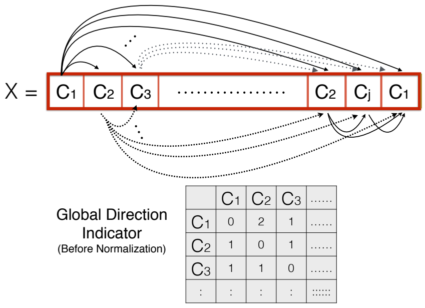
$$\begin{aligned} &F_{\textbf{Windows}}(c_a \to c_b) \\ &= \frac{\text{The number of windows where we observe } c_a \text{ and } c_b \text{ together}}{\text{The number of windows where we observe } c_b} \\ &= \frac{|\{W_i \in \textbf{Windows} \mid c_a, c_b \in W_i\}|}{|\{W_i \in \textbf{Windows} \mid c_b \in W_i\}|} \end{aligned} \quad (6)$$

We choose to accumulate the bridge weight with the sliding windows weight because these methods complement each other. Some edges that captured by the sliding windows method have zero bridging score and vice versa. Multiplying these two components instead of accumulating them would eliminate their effect in capturing inter- and intra-lecture prerequisite edges as the value of these edges will be zero.

3.1.3 First Lecture Indicator

The third component of the BEM for edge weights comes from the intuition that the context (other observed concepts) in which a new concept c_b is first introduced plays a strong role in determining what the prerequisite concepts of c_b are. We will assume that c_b is first introduced in lecture j when it has the highest term frequency of the concept c_b compared to other lectures. We call j the lecture indicator of c_b and denote it by $LI(c_b)$. When concept c_a appears in the lecture indicator of c_b ($c_a \in L_{LI(c_b)}$), then c_a might be a prerequisite to c_b. Another condition we need to examine is the temporal order of the lecture indicator of concept c_a. Naturally, when the instructor discusses a new concept, he or she needs to explain its prerequisite concepts beforehand, either in earlier lectures or in the same lecture where the new concept is being introduced. More formally, then, $LI(c_a) \le LI(c_b)$. Thus when calculating $W(c_a \to c_b)$ we consider the first lecture indicator variable FLI_{c_a,c_b} where:

$$FLI_{c_a,c_b} = \begin{cases} 1, & \text{if } c_a \in L_{LI(c_b)} \text{ and } LI(c_a) \le LI(c_b) \\ 0, & \text{otherwise} \end{cases}$$



	C1	C2	C3	......
C1	0	2	1	......
C2	1	0	1	......
C3	1	1	0	......
:	:	:	:	:::::

Figure 3: *A visualizing explanation of the Global Direction Indicator. X represents the course. The matrix contains the Global Direction Indicator (Before the normalization). Each element in the matrix represents how many times the concept c_{row} appears before the concept c_{column} in the whole entire course.*

The BEM for edge weights now becomes:

$$\begin{aligned} W(c_a \to c_b) &\approx F_{\textbf{Bridges}}(c_a \to c_b) \\ &+ F_{\textbf{Windows}}(c_a \to c_b) + FLI_{c_a,c_b} \end{aligned} \quad (7)$$

3.2 Global Direction Measure

The Global Direction Measure (GDM) is an alternative measure we propose to capture the dependency relationships between course concepts by incorporating time directly to consider not only the time ordering within lectures but also globally throughout the course delivery. In the Bridge Ensemble Measure, one problem with the sliding windows method is that the temporal order of concepts *within* a window W_i is ignored. This seems reasonable since in a single window, the instructor might mention the dependent concept before the prerequisite concepts. However, utilizing the temporal order of concepts in the entire course might improve the inference of the direction of the dependency relation. Thus, we propose the idea of the Global Direction Indicator (GDI).

The global direction indicator keeps track of the global temporal order frequency of concepts discussed in the course. In other words, it captures how many times concept c_a appears before concept c_b in the whole entire course. The more the concept c_a appears before the concept c_b, the more likelihood that the direction of the prerequisite relation is from c_a to c_b ($c_a \to c_b$). To capture the global direction indicator, we represent the course X as an ordered list of concepts discussed in all course lectures: $\vec{X} = [c_{11}, c_{12}, ..., c_{ij}, ..., c_{M1}, c_{M2}, ...]$ where i is the lecture number, j is the concept number, and M is the total number of lectures. Then, we keep track of temporal order frequency between any pair of concepts in the whole entire course. Figure 3 depicts the idea of the global direction indicator.

The formula of the global direction indicator is as follow:

$$GDI(c_a, c_b) = \frac{TOF(c_a \to c_b)}{\sum_{c_i \in C_X} TOF(c_a \to c_i)} \quad (8)$$

where TOF is the temporal order frequency, c_i are all concepts appear after c_a in the course progression. We normalize the TOF of $c_a \to c_b$ by the total number of times c_a appears before any other concept in the course to reduce the impact of popular concepts that tend to appear before almost every other concept in the course.

In addition to the global direction indicator, we modify the sliding windows method to consider the local temporal order of concepts within a single window:

$$
\begin{aligned}
&F_{\textbf{Dir-Windows}}(c_a \rightarrow c_b) \\
&\quad= \frac{\text{The number of windows where we observe } c_a \rightarrow c_b}{\text{The number of windows where we observe } c_b} \\
&\qquad= \frac{|\{W_i \in \textbf{Windows} \mid c_a \rightarrow c_b \in W_i\}|}{|\{W_i \in \textbf{Windows} \mid c_b \in W_i\}|} \qquad (9)
\end{aligned}
$$

In this case, the directed sliding windows (Dir-Windows) method captures not only the proximity of pair of concepts but also the local direction within lectures while the global direction indicator captures the frequency of the global direction.

The edge weight function according to the GDM is as follow:

$$W(c_a \rightarrow c_b) \approx GDI(c_a, c_b) \times F_{\textbf{Dir-Windows}}(c_a \rightarrow c_b) \qquad (10)$$

The rationale behind combining the GDM Components by multiplying them instead of accumulating them is to use the global direction indicator to improve the direction of edges predicted by the directed windows instead of predicting the existence of edges. The problem with the global direction indicator in predicting the edge existence is that it might give high weight to concepts that appear very often with the same direction order even if they do not appear together in any lecture.

4. EVALUATION

In this section, we demonstrate the evaluation process conducted to assess the performance of the proposed measures. We utilize the course "Text Retrieval and Search Engines"[1] to construct the concept dependency graph to evaluate our developed measures.

4.1 Building the Course Concept Space

The focus of our work is on understanding how to infer the dependency relationship between concepts, but in order to evaluate the proposed measures, we must first construct a set of concepts. There is a wide body of work which attempts to solve the problem of defining and inferring concepts [3, 9, 10]. In this paper, we use a pre-trained part-of-speech-guided phrasal segmentation, called Autophrase [10, 8], to extract salient phrases from lectures' transcripts. While Autophrase generates many good salient phrases, some phrases are either too general or are verb phrases. Our approach to improve the quality of the selected phrases is to extract phrases from weekly overviews using the same phrasal segmentation method. At the beginning of each week in the course, there is a week overview page that explains the goals and objectives of that week along with the key phrases and concepts that students need to understand. Utilizing the overview page of each week aids in filtering out meaningless phrases.

After extracting salient phrases, we manually group synonym phrases together to construct a concept. We follow Siddiqui et al. [11] definition of concepts by defining a concept as a set of salient phrases that describe it. This design decision was made to allow for flexibility in concept description since the same concept can be referred to using different phrases by different people.

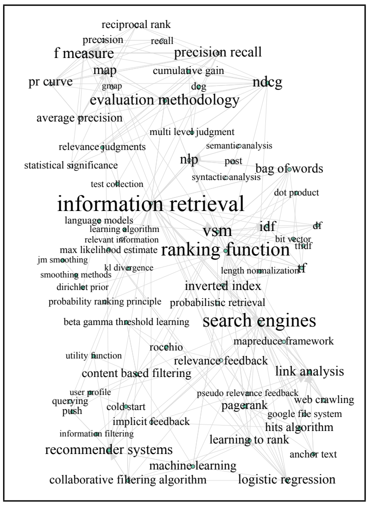


Figure 4: *The visualization of the ground truth graph generated from students data.*

4.2 Ground Truth

To evaluate the effectiveness of the proposed measures, we form a ground truth concept graph by leveraging students submissions about concept dependencies in a course (CS 410) at UIUC offered in the Spring 2017 semester that follows Coursera's "Text Retrieval and Search Engines." Students were asked to submit a weekly summary of new concepts they have learned along with prerequisite concepts. The following is an example of a student entry from week 3:

```
# f-measure: precision, recall
# pr curve: precision, recall
# map: arithmetic mean of average precision
# gmap: geometric mean of average precision
```

The total number of edges in the ground truth were 239 edges for 74 concepts in the concept space. Figure 4 visualizes the ground truth concept graph to see how concepts are related. It is clear that concepts such as "information retrieval", "search engines", "ranking function", and "evaluation methodology" have higher degree as these concepts are connected with many other concepts in the course. This is reasonable as these concepts considered fundamental in this course. Such a figure can also be seen as a useful topic map that can facilitate students browsing into course materials covering different topics flexibly; however, the map shown in this figure was constructed based on student submissions—with the proposed methods, we can construct such a map automatically for any course.

4.3 Baseline Approach

Since the problem formulation of using only transcripts to predict concept dependency is novel, strictly speaking, no previous method can be directly used to produce the desired output. The closest work that we can compare with is the work of Chaplot and Koedinger [2], which also only uses course content without any external knowledge. In their paper, they develop two methods: a text-based method called the *overlap method*, and a performance-based method. Since our work is a text-based method, we compare our measures to the overlap method. The main difference between our work and the overlap method is that we exploit the temporal features of course delivery while the overlap method does not; this makes the overlap method an ideal baseline to study the effect of the temporal features on the accuracy of edge prediction.

The overlap method, however, only predicts the prerequisite relations

[1] https://www.coursera.org/learn/text-retrieval

Table 1: *Performance (area under ROC curve) of concept graph generation for the three methods considered. Both of the new measures introduced in the paper outperform the state-of-the-art ExtendedOverlap method on both edge existence and edge direction tasks.*

	AUC (ROC)	
Method	Existence	Direction
Bridge Ensemble Measure	**0.80**	**0.81**
Global Direction Measure	**0.80**	**0.78**
ExtendedOverlap Method	0.74	0.74

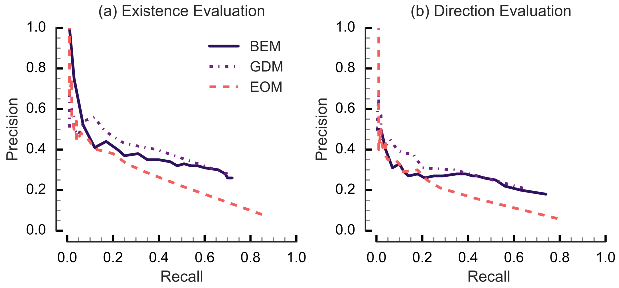


Figure 5: *The Precision/Recall Curves of Bridge Ensemble Measure (BEM), Global Direction Measure (GDM), and the baseline ExtendedOverlap Method (EOM). GDM and BEM outperform the baseline method (EOM) in both the existence evaluation and direction evaluation.*

between units (e.g. lectures) using the text overlap between units. Thus the method cannot be used directly to predict dependency between concepts, the problem that we attempt to solve. Therefore, we propose an extension called **ExtendedOverlap** for solving our problem as a baseline for comparison. Our main idea for extending the overlap method is to first map a course to a set of lectures where the concept occurred and then leverage the lecture dependency relations predicted using the overlap method to assess the dependency between two concepts by accumulating the weight of the dependency relations of lectures they belong to. All weights are normalized to be between zero and one. We implemented the overlap method using the noun phrases with document frequency normalization since they achieve the highest performance [2].

4.4 Concept Graph Performance

We conduct the evaluation of the performance of the generated concept graphs over two dimensions: edge existence and edge direction. Edge existence evaluates whether the method predicts correct edges or not while edge direction evaluation ensures not only the correctness of the edge prediction but also their direction. The AUC values of all the methods are shown in Table 1. We can notice that both the Bridge Ensemble Methods (BEM) and Global Direction Measure (GDM) outperform the baseline ExtendedOverlap (EOM) in terms of the AUC values for both the existence task and the direction task.

We also use the precision/recall curve to compare various methods as shown in Figure 5. It appears that the Global Direction Measure has the highest curve followed by the Bridge Ensemble Measure in both dimensions. This indicates that for various recall values our measures predict more accurate edges than the baseline. It is also interesting to notice that in the precision/recall curve of the existence evaluation (Figure 5 (a)), the Bridge Ensemble Measure has the highest precision when the recall is less than **0.1** while in the precision/recall curve of the direction evaluation (Figure 5 (b)) has the lowest precision until it reaches the recall value of **0.2**. This indicates that, in the interval **[0.0,0.2]**, the Bridge Ensemble method captures the existence of the edges but fails at specifying the correct direction. To examine the reason, we study the performance of various components of the Bridge Ensemble Measure as depicted in Figure 6. It is appear that the undirected sliding windows method has the highest curve in the existence evaluation (Figure 6 (a)) and since it only captures the proximity of pair of concepts and how they are related, it surges the precision/recall curve of the existence performance in the interval **[0.0,0.2]** by capturing correct prerequisite edges. However, since the temporal feature is only used in limited way as a binary variable among lectures through bridges and first lecture indicator components, it sometimes fails at predicting the correct direction of edges between concepts that only appears within the same lectures. In contrast, the Global Direction Measure exploits the global direction indicator that keeps track of the global temporal order frequency and hence emphasizes or corrects the direction captured by the directed sliding windows method as depicted in Figure 7. It is clear from Figure 7 that the global direction indicator improves the edge direction of the directed windows method when the recall value is less than **0.2** while it emphasize the edge direction of the directed windows after that.

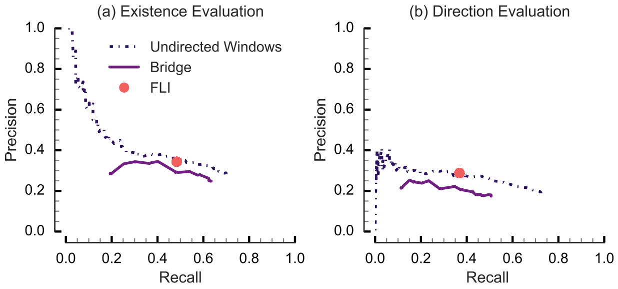


Figure 6: *The comparison between the performance of the Bridge Ensemble Measure components. While the undirected sliding windows correctly captured the edge existence in the interval [0.0, 0.2], it fails at predicting edge directions.*

To further analyze the differences between the Bridge Ensemble Measure and the Global Direction Measure, we examine their behavior in the existence dimension. We found that all true positive edges and false positive edges captured by Global Direction measure are also captured by Bridge Ensemble Measure. However, Bridge Ensemble Measure has more false positive edges (59 edges) and more true positive edges (only 4 edges). We examine the source of the extra false positive edges in the Bridge Ensemble Measure and found that 73% came from the bridge method, 3% came from the first lecture indicator, and 22% are from both the bridge method and the first lecture indicator while the sliding windows has zero contribution (0%). Further examination of these extra false positive errors shows that some of them capture long distance dependencies such as the relation "natural language processing" → "recommender systems", which captures the dependency between the concepts explained in the first and last lectures. By examining the source of this relation, we found that the bridge method makes the inference of the relation. As

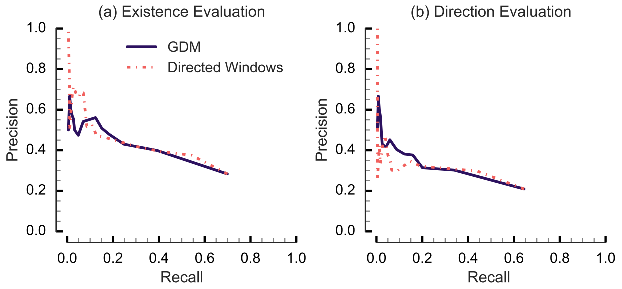


Figure 7: *The effect of the global direction indicator on the Global Direction Measure. The GDI improves the edge direction of the directed windows method when the recall value is less than **0.2** while it emphasize the edge direction of the directed windows after that*

mentioned earlier, bridge method captures the dependency relations between concepts across lectures and, in contrast to the sliding windows method, it does not require the proximity of concepts within lectures' transcripts. This property of the bridge method gives the Bridge Ensemble Measure the ability to capture long distance relations between concepts in contrast to the Global Direction Measure which only captures the local dependencies between concepts (within lectures).

We also conduct a qualitative analysis of the false positive edges to examine the reason of the high values and hence the low precision values. We found three types of false positive edges that we may actually consider correct relations. First, the transitive property edges that are captured by our measures are not always specified in the ground truth edges. For example, students specify the relations "length normalization" → "ranking function", and "ranking function" → "vector space model". While both our measures and the baseline capture these relations, they go further and also capture the transitive relation "length normalization" → "vector space model". Second, there are issues with relations with differing concept granularities. For instance, students specify a dependency relation "language models" → "dirichlet prior smoothing" while the generated graphs by the three methods capture the relation "language models" → "smoothing methods." The concept "smoothing methods" is more general than the concept "dirichlet prior smoothing." Third, there are missing "true" relations that the students did not specify in the ground truth. For example, students did not specify the following relations that are captured by our measures: "tfidf" → "bm25", and "length normalization" → "bm25." In general, the three types of false positive errors can justify to some extent the high values of the false positive errors and thus the low values of the precision.

In general, the Bridge Ensemble Measure and the Global Direction Measure outperform the baseline in terms of AUC and precision/recall curves, with the Global Direction Measure having the overall highest performance. These results emphasize the positive effect of the temporal feature on improving the accuracy of the generated concept graph.

5. LIMITATIONS

There are some limitations in our study. First, in the evaluation we have not examined the robustness of our measures compared to the baseline utilizing other courses taught by different instructors. Second, we use the students' perspectives of the concept dependency graph as a ground truth, and we are the first study to do so. However, in the future we plan to compare various methods' performance by utilizing not only the students' perspectives of the concept graph but also one generated by instructors. Third, in this study, we include an edge in the ground truth even if only one student specifies it; in the future we plan to use some agreement measures before including an edge in the ground truth. Fourth, we represent the course concept graph according to the dependency structure without distinguishing whether the dependency relation captures the hierarchical structure or real prerequisite relationships. We believe that the ideal structure of the concept dependency graph is a hierarchical graph with cross link edges where the hierarchical structure captures the "general concept" to "specific concept" relations while the cross links depict the prerequisite relationships between concepts.

6. RELATED WORK

Most prior work focuses on relationships between concepts such as similarity relations [13] and hierarchical relations [5]. Although the most important concept relation to learners is the dependency or prerequisite relation, this relation has been the least studied [4]. Some prior works utilize Wikipedia articles [6, 12, 1, 7], scientific corpora [4], or educational materials from online educational platforms [14, 2, 7] to model the dependency structure between concepts. While many studies utilized external knowledge to recover the prerequisite relations [14, 7] , Chaplot and Koedinger [2] utilize the course content with students performance to infer such relation. In contrast, to make our method more accessible, we exploit only the easily accessible educational materials to model the dependency relations among course concepts.

Previous research represents graph concepts in various ways. Gordon et al. [4] identify concepts using LDA topic modeling that fails in identifying finer-grained concepts. Yang et al. [14] explored four different representations and found that word and category representations have similar performance; however, word representation has slightly better performance on some data sets. One problem with using category representations is that mapping phrases to Wikipedia categories affects concept granularities by preferring more general concepts. On the other hand, Chaplot and Koedinger [2] found that noun phrase representation outperforms other representations. Therefore, in this study, we utilize noun phrase representation but extend it using temporal information.

Previous work developed supervised [1, 12, 14] and unsupervised approaches [6, 7, 2] to predict the dependency relationships among concepts. Several studies rely on external knowledge to predict prerequisite relations across courses [14, 7] while we only leverage course materials to model the dependency relations within a course not between courses. Chaplot and Koedinger [2] address the dependency structure within courses, but between units instead of concepts taught within units. Another main difference is the use of the temporal feature in the course delivery to model the dependency structure as we are the first study that exploits the temporal feature.

7. CONCLUSIONS

In this paper, we leverage the accessible MOOC content and incorporate the temporal feature of the course to construct a concept dependency graph. We developed Bridge Ensemble Measure and Global Direction Measure that exploit the temporal order in course delivery to model the dependency structure. We revealed in the evaluation that both developed measures outperform the baseline method in AUC and in precision recall curves. This finding emphasizes the positive effect of utilizing the temporal feature of course progression.

8. REFERENCES

[1] R. Agrawal, B. Golshan, and E. Papalexakis. Toward data-driven design of educational courses: A feasibility study. *JEDM-Journal of Educational Data Mining*, 8(1):1–21, 2016.

[2] D. Chaplot and K. R. Koedinger. Data-driven automated induction of prerequisite structure graphs. In *Proceedings of the 9th International Conference on Educational Data Mining*, pages 318–323. EDM, 2016.

[3] A. El-Kishky, Y. Song, C. Wang, C. R. Voss, and J. Han. Scalable topical phrase mining from text corpora. *Proceedings of the VLDB Endowment*, 8(3):305–316, 2014.

[4] J. Gordon, L. Zhu, A. Galstyan, P. Natarajan, and G. Burns. Modeling concept dependencies in a scientific corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 866–875, 2016.

[5] I. Jonyer, D. J. Cook, and L. B. Holder. Graph-based hierarchical conceptual clustering. *Journal of Machine Learning Research*, 2(Oct):19–43, 2001.

[6] C. Liang, Z. Wu, W. Huang, and C. L. Giles. Measuring prerequisite relations among concepts. In *EMNLP*, pages 1668–1674, 2015.

[7] C. Liang, J. Ye, Z. Wu, B. Pursel, and C. L. Giles. Recovering concept prerequisite relations from university course dependencies. In *AAAI*, pages 4786–4791, 2017.

[8] J. Liu, L. Jiang, Z. Wu, Q. Zheng, and Y. Qian. Mining learning-dependency between knowledge units from text. *The VLDB Journal-The International Journal on Very Large Data Bases*, 20(3):335–345, 2011.

[9] J. Liu, J. Shang, C. Wang, X. Ren, and J. Han. Mining quality phrases from massive text corpora. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1729–1744. ACM, 2015.

[10] J. Shang, J. Liu, M. Jiang, X. Ren, C. R. Voss, and J. Han. Automated phrase mining from massive text corpora. *arXiv preprint arXiv:1702.04457*, 2017.

[11] T. Siddiqui, X. Ren, A. Parameswaran, and J. Han. Facetgist: Collective extraction of document facets in large technical corpora. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 871–880. ACM, 2016.

[12] 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*, pages 307–315. Association for Computational Linguistics, 2012.

[13] R. W. White and J. M. Jose. A study of topic similarity measures. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 520–521. ACM, 2004.

[14] Y. Yang, H. Liu, J. Carbonell, and W. Ma. Concept graph learning from educational data. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 159–168. ACM, 2015.

Predicting Individualized Learner Models Across Tutor Lessons

Michael Eagle, Albert Corbett, John Stamper, Bruce Mclaren
Human-Computer Interaction Institute
Carnegie Mellon University
{meagle,ac21,jstamper,bmclaren}@andrew.cmu.edu

ABSTRACT

In this work we use prior to tutor-session data to generate an individualized student knowledge model. Intelligent learning environments use student models to individualize curriculum sequencing and help messages. Researchers decompose the learning tasks into sets of Knowledge Components (KCs) that represent individual units of knowledge; the student model estimates a parameters for each KC, but not for each student. Using existing performance data to adjust parameters for each individual student improves model fit, and leads to different practice recommendations. However, in order to be implemented in a live system we need to have a method to estimate the student parameters using only the student‘s prior activities. In this work, we use data collected from student reading, prior tutor lessons, to predict individualized difference weights for parameters of a Bayesian Knowledge Tracing (BKT) variant. We find that best-fitting student parameters trained on previous lessons do not directly transfer to new lessons; however, we can effectively predict the student parameters for the new lesson by using features derived from prior lessons, and prior to tutor text-reading transaction data.

KEYWORDS

Individualization, Student Modeling, BKT, Genetics

1 INTRODUCTION

Learner models of domain knowledge have been successfully employed for decades in intelligent tutoring systems (ITS), to individualize both curriculum sequencing [8, 19, 23, 24] and help messages [6, 13]. Bayesian methods are frequently employed in ITSs to infer student knowledge from performance accuracy, as in the citations above, as well as in other types of learning environments [21], and Bayesian modeling systems have been shown to accurately predict students‘ tutor and/or posttest performance [7, 8, 14, 24]. These models generally individualize modeling parameters for individual knowledge components (KCs, also referred to as skills) [16], but not for individual students. Several studies have shown that individualizing parameters for students, as well as for KCs, improves the quality of the models [7, 18, 22, 27]. These approaches to modeling individual differences among students have monitored student performance after the fact, in tutor logs that have been previously collected to derive individualized student parameters for the tutor module(s). While these efforts have proven successful, they don‘t achieve the goal of dynamic student modeling within an ITS, since estimating and using individualized parameters concurrently within a tutor lesson is quite difficult. In this paper we examine how well individual differences in student learning in a lesson of the Genetics Cognitive Tutor [7] can be predicted ahead of time from two types of prior online activities: reading instructional text and solving problems in prior tutor lessons. In the following sections we describe Knowledge Tracing, the on-line student activities, the predictors derived from students‘ reading and prior tutor activities, and our success in using these predictors to model individual differences in the tutor.

1.1 Modeling Framework

Bayesian Knowledge Tracing (BKT) estimates the probability that a student knows each of the knowledge components (KC) in a tutor lesson. It employs a two-state Bayesian learning model — at any time a student either knows or does not know a given KC — and employs four parameters, which are estimated separately for each KC: $p(L_0)$ — initial knowledge the probability a student has learned how to apply a KC prior to the first opportunity to apply it in a lesson. $p(T)$ — learning rate the probability a student learns a KC at each chance to apply it. $p(G)$ — guessing the probability a student will guess correctly if the KC is not learned. $p(S)$ — slips the probability a student will make an error when the KC has been learned. BKT is employed in Cognitive Tutors to implement Cognitive Mastery, in which the curriculum is individualized to assign only the number of practice opportunities needed to enable the student to "master" each of the KCs, which is generally operationalized as a 0.95 probability that the student has learned the KC.

1.1.1 Individual Differences. Knowledge Tracing and Cognitive Mastery generally employ best-fitting estimates of each of the four parameters for each individual KC but not for individual students. In this work, we incorporate individual differences among students into the model in the form of individual difference weights. Following Corbett and Anderson [8], four best-fitting weights are estimated for each student, one weight for each of the four parameter types, wL0, wT, wG, wS. In estimating and employing these individual difference weights (IDWs), we convert each of the four probability estimates to odds form (p/(1-p)), multiply the odds by the corresponding student-specific weight and convert the resulting odds back to a probability. (See [8] for computational details.)

In this paper we focus on four types of BKT models for the third lesson in a Genetics Cognitive Tutor curriculum on genetic pathways analysis to examine how well IDWs in a tutor lesson can be predicted from prior online activities. The four models are: (1) a standard BKT model (SBKT) with no individualization, (2) a model with best-fitting IDWs for lesson 3 (BFIDW-L3), (3) models with best-fitting IDWs from prior lessons, and (4) a model with predicted individual difference weights derived from earlier activities. We compare how much each of the three types of individualized models improves upon the non-individualized SBKT fit (1).

EDM‘18, July 2018, El Buffalo, New York USA
.

Eagle et al. [11] estimated individual difference weights using reading performance data, pretest scores, resulting in a predictive model 40% as effective as the best-fitting model; the predictive model was improved for a second lesson reaching 60% of the best-fitting model by using previous lesson data [11]. As pretests do not necessarily appear in all online environments, in this paper, we examine how well we can predict IDWs in a third lesson with the same types of reading measures as in [11, 12] along with an expanded set of tutor performance measures.

2 STUDENT ACTIVITIES IN THIS STUDY

The students in this study worked through two successive topics in the genetic pathway analysis curriculum within the Genetics Cognitive Tutor. The first topic, gene interaction, examines the different ways two genes can interact in controlling a single trait. e.g., coat color in cattle. The second topic, gene regulation, focuses on three-gene systems in which two genes function together to control the expression of the third gene.

For each topic students completed five activities: reading instructional text, taking a conceptual-knowledge pretest, completing two Genetics Cognitive Tutor lessons and completing a problem-solving posttest. The two tutor lessons for each topic require students to think about the topic in contrasting ways. In the first, “forward reasoning“ or process modeling lesson, students are given descriptions of how genes interact in a system and reason about the resulting behavior of the system. In the second, “backward reasoning,“ or abductive reasoning lesson, students are given descriptions of how genetic systems behave, and draw conclusions about how the underlying genes interact.

Online Instructional Text: The first text on gene interaction consists of 23 screens, and the second gene regulation text consists of 20 screens. The screens are structured like pages in a book. Students can move forward and backward through the screens, one screen at a time. After a student touches each page once a “done“ button appears and the student can then continue reading, or exit at any time.

Cognitive Tutor Lessons: The first tutor lesson, Gene Interaction Process Modeling, consists of 5 problems, averaging 45 steps per problem. The second tutor lesson, Gene Interaction Abductive Reasoning, consists of 6 problems, averaging 25 steps each. Features of student performance in these two lessons (along with features of their reading performance) are employed to predict individual differences in the third tutor lesson, Gene Regulation Process Modeling, which consists of 9 problems with 27 steps each.

3 PREDICTORS

In this study, we examine three types of student performance variables as predictors of best fitting Lesson 3 IDWs: Aspects of reading the two texts, Lesson 1 and Lesson 2 IDWs, and features of student performance in completing tutor Lessons 1 and 2.

3.1 Instructional Text Reading Predictors

Two types of measures of students‘ reading performance were derived for both the Topic 1 (gene interaction) and Topic 2 (gene regulation) instructional texts: reading time per page and pages revisited in the text. Eagle et al [11, 12] found that both types of reading measures for the gene interaction text entered reliably into predictive models for IDWs for both of the gene interaction tutor lessons.

Reading Time: A factor analysis was performed on log reading times for the 23 Topic 1 pages and a factor analysis on log reading times for the 20 Topic 2 pages to reduce the number of predictors. Each analysis yielded (a different set of) four reading time factors.

Text Pages Revisited: Students may choose to strictly read forward through a text, or may choose to revisit earlier pages. Two measures of student behavior in revisiting text pages were calculated: the number of pages re-read and the number of intervening pages traversed in re-reading text pages.

3.2 Prior Lesson Model Predictors

We derived a total of total of 16 predictors from the lesson 1 and 2 student models.

Individual Difference Weights: Three sets of best-fitting individual-difference weights were derived (1) for the 31 KCs in Lesson 1, (2) for the 22 KCs in Lesson 2, and (3) for the combined set of 53 KCs in Lessons 1 and 2.

Probabilities students learned the Lesson 1 & 2 KCs: At the end of a lesson, BKT yields a probability that a student knows each KC in the lesson. Two measures of each student‘s knowledge at the end of each lesson were calculated: the number of unmastered skills and the minimum probability the student knows any single KC.

3.3 Tutor Performance Features

Finally, thirteen predictors based on student performance in each of the two tutor lessons were derived. Raw error rate for students‘ first action at each problem- solving step in each lesson, and average response time for students‘ first action at each problem-solving step in each lesson were calculated.

In addition, for each of the two lessons the following 11 measures of students‘ metacognitive skills were calculated. Most of these have previously been shown [10] to correlate with measures of robust learning, including direct transfer of knowledge, which is similar students‘ initial knowledge, pL0, and preparation for future learning, which is similar to students; learning rate wT:

Help avoidance [1]: the proportion of problem solving steps in which the probability the student knows the relevant KC is low and the student‘s first action is an error instead of a hint request.

Bug Messages: the proportion of each student‘s actions in which a bug message (an error message generated when a student‘s behavior matches a known misconception) is followed by a long pause, and the proportion in which a bug message is followed by a short pause.

Hint Messages: the proportion of each student‘s actions in which a hint request is followed by a long pause, and the proportion in which a hint request is followed by a short pause.

Known-KCs: the proportion of each student‘s actions in which the student knows the relevant skill well and there is a long pause before responding, and the proportion in which the student knows the skill well and there is a short pause.

Off-Task and Gaming Variables: The proportion of actions in which an automatic detector determined the student was gaming the system [9] was calculated, (e.g., systematic guessing, or quickly drilling down through the tutor‘s hints to find the correct answer), as was the proportion of fast responses that were not identified as gaming by the detector. Also, we calculated for each student both the proportion of actions in which an automatic detector determined the student was off task [3] and the proportion of actions where there was a long pause not identified as off-task.

4 METHODS AND MATERIALS

The data analyzed in this study come from 80 CMU undergraduates enrolled in either genetics or introductory biology courses who were recruited to participate in this study for pay. The students participated in two 2.5-hour sessions on consecutive days in a campus computer lab. The first session focused on the first topic, gene interaction and the second session focused on the second topic, gene regulation. In each session students completed five activities: Read an on-line instructional text on the session topic; completed a pretest on the topic; completed two Genetics Cognitive Tutor modules on the session topic, a “forward“ process-modeling module and a “backward“ abductive reasoning module; and completed a problem-solving posttest. This study focuses on modeling the 22,681 problem-solving steps in the third, gene regulation process-modeling tutor lesson.

4.1 Fitting Procedures

We first found best-fitting group parameter estimates for each of the 4 parameters (pL0, pT, pG, pS) in the standard BKT (SBKT) model for each of the 47 KCs in Lesson 3, with nonlinear optimization. We optimize on negative log-likelihood and generate the best fitting set of group parameters for each of the 47 KCs. Both pG and pS were bounded to be less than 0.5, as in Baker et al., [4] to avoid paradoxical results that arise when these performance parameters exceed 0.5 (e.g., a student with a higher probability of knowing a KC is less likely to apply it correctly.)

Second, we generate individualized BKT models by optimizing a new set of four Individual Difference Weights (IDWs,) one for each of the four standard BKT parameters, wL0, wT, wG, wS, for each of the 80 students. The optimization process takes as input the SBKT model, and the observed student opportunities, and produces the best fitting set of IDWs for each student.

Third, we derived the 6 reading features for text 2, and tutor performance measures for Lesson 1 and 2 that had not previously been derived in [11, 12]. Along with the measures from text 1, the best-fitting IDWs for Lessons 1 and 2, and the Lesson-1 measures that had been derived previously [11, 12], this yields a total set of 50 predictor variables.

We employed these 12 reading variables (6 for each topic) and the 38 tutor performance variables (19 for each lesson) to independently predict the four Lesson 3 IDWs: wL0, wT, wG, wS. Since we are predicting multiplicative weights, we fit a transformation of the weights w/(1+w). This transformation has the property that the neutral weight 0.5 (which does not modify the corresponding best-fitting group parameter), is the midpoint of the transformed scale.

4.2 Model and Feature Selection

In order to generate the predictive IDW model we first reduced the number of features with Least Angle Regression (LAR) [25] a variant of Lasso. For each of the four Lesson 3 IDWs we use LAR to select the best 12 predictors (out of 50,) Twelve predictors were selected to match with models presented in work by Eagle et al., [11, 12].

Table 1: Goodness of fit for Lesson 3 tutor performance.

Model	RMSE	Accuracy
SBKT	0.399	0.765
BFIDW-L3	0.368	0.806
BFIDW-L1	0.4	0.766
BFIDW-L2	0.394	0.774
BFIDW-L12	0.389	0.778
PrIDW-L12	0.38	0.791

We then built a robust regression model with the 12 predictors for each of the IDWs. Robust regression is less sensitive to outliers, variable normality, and other violations of standard linear regression assumptions [2]. In order to control for the false discovery rate, we adjusted for multiple comparisons in the coefficient significance tests [5].

Finally, we employed the standard BKT model for lesson 3, the best fitting IDWs from each of the three lessons, and the various sets of predictor variables to generate 5 new IDW BKT models for Lesson 3, yielding a total of six BKT model variants displayed below. Analysis work was performed using R [15], Optimx [20], rlm [26], and lars [25].

Six BKT models calculated in this analysis for Lesson 3:

SBKT: Standard BKT non-individualized model with best-fitting group parameter estimates

BFIDW-L3: Individualized BKT model with best-fitting IDWs for Lesson 3

BFIDW-L1: Individualized BKT model with best-fitting IDWs for KCs in Lesson 1

BFIDW-L2: Individualized BKT model with best-fitting IDWs for KCs in Lesson 2

BFIDW-L12: Individualized BKT model with best-fitting IDWs for KCs in both Lessons 1 & 2

PrIDW-L12: Individualized BKT with predicted IDWs from reading and from Lesson 1 and Lesson 2 tutor performance features.

5 RESULTS AND DISCUSSION

Table 2 displays the overall fit to students‘ Lesson 3 tutor performance of the six models. Column 2 displays root mean squared error (RMSE) for the fits and column 3 displays Accuracy (the probability a model correctly predicts students‘ correct or incorrect responses with a 0.5 threshold on predicted accuracy).

Best-fitting IDWs for Lesson 3. The RMSE for the SBKT model with best fitting Lesson 3 parameter estimates, but no individualization is 0.399, as displayed in row 1. The remaining five rows display the five individualized models. BFIDW-L3 in row 2 employs best-fitting IDWs derived from the lesson 3 data. This model necessarily yields the best fit; it improves the goodness of fit by 7.8% over the SBKT model, reducing RMSE from 0.399 to 0.368.

Direct transfer of IDWs from Lessons 1 and 2. The next 3 rows display goodness of fit when the best fitting IDWs from Lesson 1, from Lesson 2, and from Lessons 1 & 2 combined, are employed directly in modeling Lesson 3 performance. As can be seen, BFIDW-L1, with IDWs from Lesson 1, and BFIDW-L2 with IDWs from Lesson 2 have little impact on the overall goodness of fit compared to SBKT, changing RMSE -0.03% and 1.6% respectively. BFIDW-L12 with refitted IDWs for the 53 KCs in both lessons has a slightly larger effect, improving on the SBKT fit by 3.2% reducing it to 0.394.

Predicted IDWs based on reading and Lessons 1 and 2 performance. The last row in the table displays RMSE for the PrIDW-L12 model in which reading measures from both texts and tutor performance measures from lessons 1 and 2 are employed to predict Lesson 3 IDWs. This model reduces RMSE to 0.380; it is about 60% as successful as the best-fitting BFIDW-L3 in reducing RMSE (and twice as successful as BFIDW-12).

Individualization and Mastery. Small differences in model fits can have large effects on the amount of practice assigned to students [11, 12, 17]. Following [11, 12], we calculated the approximate amount of practice that would be necessary for students to reach mastery under each of the six models in Table 2, and found general agreement among the five IDW models compared to the standard SBKT model. On average 51 students would have needed less practice under any of the 5 IDW models than under the SBKT model (range 46-57) and on average they would have required 54 fewer practice opportunities across all the lesson-3 KCs (range 42-64). On average 29 students would have needed more practice (range 22-30) and they would have needed an average of 23 more opportunities across all KCs (range 18-23). We take BFIDW-L3 (with best fitting Lesson-3 IDWs) as the gold standard in this comparison, and while the PrIDW-L3 model fits the lesson 3 data better than BFIDW-L12, the latter model agrees slightly better with BFIDW-L3 than does PrIDW-L3 (94% vs 91%). More work is needed to understand the relationship between model fit and mastery recommendations, but the general agreement between the IDW models suggests that a variety of evidenced-based IDW sets can improve efficiency in guiding students to mastery, compared to the SBKT model.

Table 2: Coefficient Summary Table

Pred.	wL0	wT	wG	wS
(Inter.)	1.012***	0.866***	0.242	0.306*
RT	T1F1[1]0.63	T2F3 0.043	T1F1 -0.034	T1F4 -0.034*
RT	T1F3 -0.066		T1F4 0.060	T2F1 -0.025
RT			T2F3 -0.039	
RT			T2F1 -0.017	
Pg re.				
Pg dist.				
wL0			L1 0.106	
wT		L2 0.171	L2 -0.080	
wG	L1[2]0.095		L2 0.034	L2 0.026
wS	L1 -0.235 L2 -0.239	L1 -0.433***		L2 0.214
Min. pLn				
Mast. KC		L2 -0.006***		
Err Rate.	L2 -0.411	L2 0.068		
Mean RT	L2 0.010	L2 0.016		
Help Av	L1 -2.996	L1 -1.773		L1 1.036 L2 1.714*
Bug-LP			L2 15.672	
Bug-SP		L2 -5.514	L1 9.728	L2 -4.978
Hint-LP				
Hint-SP				
Kn-LP	L2 -0.726	L2 -0.275		L2 -0.616
Kn-SP	L2 -1.869***		L1 1.287	L2 0.386 L1 0.791
Gaming	L1 -0.107	L2 -0.851	L2 0.534	
SP-NotG				
Off-Task	L1 -1.766	L1 -4.94**	L1 2.847	L1 2.378 L2 -2.624*
LP-NotOT		L2 0.033		
RMSE	0.16	0.157	0.192	0.139

(* < 0.10, ** < 0.05, *** < 0.01)

[1] T1F1 = Topic 1 (gene interaction), Factor 1

[2] L1 = Lesson 1 wS (slip IDW)

5.1 IDW Predictive Models

Table 3 displays the coefficients for each of the predictors in the regression models for each of the four Lesson 3 IDWs. As in [11], Lasso was used to identify the best 12 predictors for each of the four IDWs. The predictors that enter reliably into the four robust regression models are highlighted with asterisks.

The predictors that enter into the four models are rather eclectic. Reading time factors from the first text are among the top 12 predictors in three of the four IDWs models, as are reading time factors for the second text. The first text is on a different topic (gene interaction) than Lesson 3 (gene regulation). This suggests the reading time factors may be tapping learning strategy rather than the specific knowledge acquired.

Among the tutor performance measures in Table 3, slightly more came from Lesson 2 than Lesson 1, 25 vs. 15, but the difference is not significant. Whereas Lesson 1 and Lesson 3 employ related reasoning strategies — “forward“ process modeling rather than “backward“ abduction, Lessons 2 and 3 are closer in time; both of these relationships may contribute to predictive effectiveness, with perhaps a slight advantage for recency.

The 19 total tutor performance variables fall into four broad types: the 4 IDWs, two BKT measures of student knowledge at the end of each lesson, two raw measures of performance, error rate and mean response time, and finally, the 11 “metacognitive“ measures, including use of help, response time in specific contexts, gaming and off-task behaviors. None of these four categories emerges as a stronger predictor than the others. Overall, each of the 19 variables enters into an average of 2.1 models, and the average number of models for the variables within any of the four categories does not depart much from this mean. Perhaps most surprisingly, the Lesson 1 and Lesson 2 IDWs are not especially strong predictors of Lesson 3 IDWs. Lesson 1 wL0 is among the top 12 predictors for just one model, Lesson 2 wT appears twice in Table 3, Lesson 1 or 2 wG appears three times, and Lesson 1 or 2 wS appears four times. The average number of models in which these variables appear, 2.5, is not much different from the overall average of 2.1.

Finally, among the 11 metacognitive features, Lesson 1 off-task behavior is perhaps the strongest predictor of Lesson 3 IDWs; it appears among the top 12 variables in all four models, and is significant in one of the models.

6 CONCLUSION

This study examines methods for predicting individual difference weights for students in BKT learning parameters (intercept and rate) and performance (guess and slip) for the third lesson in a Cognitive Tutor curriculum. This is an important issue because integrating IDWs into an intelligent tutor lesson is easier if the IDWs can be assigned before the student starts working in the lesson. We evaluate the different estimated IDWs by examining how well they fit student performance in Lesson 3, compared to (1) standard SBKT with no IDWs, and (2) a model with best-fitting weights for Lesson 3.

We find that directly applying the best-fitting IDWs from either of two prior lessons in the curriculum, or from both lessons combined, does not appreciably improve goodness of fit for Lesson 3, compared to the SBKT model. In contrast, estimating lesson-3 IDWs from measures of students‘ prior reading performance, and performance in the two prior tutor lessons, is more successful; it is 60% as successful as the best-fitting Lesson-3 IDW model in improving the goodness of fit compared to the SBKT model.

Several secondary conclusions emerge. First, a prior study [12] obtained very similar success in predicting IDWs based on reading performance, pretest performance and a smaller set of tutor performance measures. This study demonstrates that IDWs can be successful predicted without including pretest measures. This is potentially important since pretests may not be available in online learning environments. Second, among reading time measures and a wide range of tutor performance measures, no category of measures emerged as an especially strong predictor of Lesson 3 IDWs; instead it appears that predictive success depends on a broad range of predictor variables. Finally, reading time measures prove to be useful predictors of students‘ problem-solving behaviors in a subsequent tutor lesson, including reading time measures for text on a topic unrelated to that tutor lesson. This suggests that the reading time measures may reflect knowledge-acquisition strategies, as well as any knowledge acquired.

REFERENCES

[1] Vincent Aleven, Ido Roll, Bruce M McLaren, and Kenneth R Koedinger. 2016. Help helps, but only so much: Research on help seeking with intelligent tutoring systems. *International Journal of Artificial Intelligence in Education* 26, 1 (2016), 205–223.

[2] Robert Andersen. 2008. *Modern Methods for Robust Regression.* SAGE Publications, Inc. https://doi.org/10.4135/9781412985109

[3] Ryan SJd Baker. 2007. Modeling and understanding students’ off-task behavior in intelligent tutoring systems. In *Proceedings of the SIGCHI conference on Human factors in computing systems.* ACM, 1059–1068.

[4] Ryan SJ Baker, Albert T Corbett, and Vincent Aleven. 2008. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *International Conference on Intelligent Tutoring Systems.* Springer, 406–415.

[5] Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)* (1995), 289–300.

[6] Cristina Conati, Abigail Gertner, and Kurt Vanlehn. 2002. *User Modeling and User-Adapted Interaction* 12, 4 (2002), 371–417. https://doi.org/10.1023/a:1021258506583

[7] Albert Corbett, Linda Kauffman, Ben Maclaren, Angela Wagner, and Elizabeth Jones. 2010. A Cognitive Tutor for Genetics Problem Solving: Learning Gains and Student Modeling. *Journal of Educational Computing Research* 42, 2 (feb 2010), 219–239. https://doi.org/10.2190/ec.42.2.e

[8] Albert T. Corbett and John R. Anderson. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modelling and User-Adapted Interaction* 4, 4 (1995), 253–278. https://doi.org/10.1007/bf01099821

[9] Ryan SJ d Baker, Albert T Corbett, Ido Roll, and Kenneth R Koedinger. 2008. Developing a generalizable detector of when students game the system. *User Modeling and User-Adapted Interaction* 18, 3 (2008), 287–314.

[10] Ryan S. J. d. Baker, Albert T. Corbett, and Sujith M. Gowda. 2013. Generalizing automated detection of the robustness of student learning in an intelligent tutor for genetics. *Journal of Educational Psychology* 105, 4 (2013), 946–956. https://doi.org/10.1037/a0033216

[11] Michael Eagle, Albert Corbett, John Stamper, Bruce M McLaren, Ryan Baker, Angela Wagner, Benjamin MacLaren, and Aaron Mitchell. 2016. Predicting Individual Differences for Learner Modeling in Intelligent Tutors from Previous Learner Activities. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization.* ACM, 55–63.

[12] Michael Eagle, Albert Corbett, John Stamper, Bruce M. McLaren, Angela Wagner, Benjamin MacLaren, and Aaron Mitchell. 2016. Estimating Individual Differences for Student Modeling in Intelligent Tutors from Reading and Pretest Data. In *Intelligent Tutoring Systems.* Springer International Publishing, 133–143. https://doi.org/10.1007/978-3-319-39583-8_13

[13] Rajaram Ganeshan, W. Lewis Johnson, Erin Shaw, and Beverly P. Wood. 2000. Tutoring Diagnostic Problem Solving. In *Intelligent Tutoring Systems.* Springer Berlin Heidelberg, 33–42. https://doi.org/10.1007/3-540-45108-0_7

[14] Yue Gong, Joseph E. Beck, and Neil T. Heffernan. 2010. Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting Procedures. In *Intelligent Tutoring Systems.* Springer Berlin Heidelberg, 35–44. https://doi.org/10.1007/978-3-642-13388-6_8

[15] Ross Ihaka and Robert Gentleman. 1996. R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics* 5, 3 (sep 1996), 299–314. https://doi.org/10.1080/10618600.1996.10474713

[16] Kenneth R. Koedinger, Albert T. Corbett, and Charles Perfetti. 2012. The Knowledge-Learning-Instruction Framework: Bridging the Science-Practice Chasm to Enhance Robust Student Learning. *Cognitive Science* 36, 5 (apr 2012), 757–798. https://doi.org/10.1111/j.1551-6709.2012.01245.x

[17] Kenneth R. Koedinger, John C. Stamper, Elizabeth A. McLaughlin, and Tristan Nixon. 2013. Using Data-Driven Discovery of Better Student Models to Improve Student Learning. In *Lecture Notes in Computer Science.* Springer Berlin Heidelberg, 421–430. https://doi.org/10.1007/978-3-642-39112-5_43

[18] Jung In Lee and Emma Brunskill. 2012. The Impact on Individualizing Student Models on Necessary Practice Opportunities. *International Educational Data Mining Society* (2012).

[19] Michael Mayo and Antonija Mitrovic. 1999. Estimating Problem Value in an Intelligent Tutoring System Using Bayesian Networks. In *Advanced Topics in Artificial Intelligence.* Springer Berlin Heidelberg, 472–473. https://doi.org/10.1007/3-540-46695-9_42

[20] John C. Nash and Ravi Varadhan. 2011. Unifying Optimization Algorithms to Aid Software System Users:optimxforR. *Journal of Statistical Software* 43, 9 (2011). https://doi.org/10.18637/jss.v043.i09

[21] Zachary Pardos, Yoav Bergner, Daniel Seaton, and David Pritchard. 2013. Adapting bayesian knowledge tracing to a massive open online course in edx. In *Educational Data Mining 2013.*

[22] Zachary A. Pardos and Neil T. Heffernan. 2010. Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. In *User Modeling, Adaptation, and Personalization.* Springer Berlin Heidelberg, 255–266. https://doi.org/10.1007/978-3-642-13470-8_24

[23] Steve Ritter, Michael Yudelson, Stephen E Fancsali, and Susan R Berman. 2016. How mastery learning works at scale. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale.* ACM, 71–79.

[24] Valerie J. Shute. 1995. SMART: Student modeling approach for responsive tutoring. *User Modelling and User-Adapted Interaction* 5, 1 (1995), 1–44. https://doi.org/10.1007/bf01101800

[25] Robert Tibshirani, Iain Johnstone, Trevor Hastie, and Bradley Efron. 2004. Least angle regression. *The Annals of Statistics* 32, 2 (apr 2004), 407–499. https://doi.org/10.1214/009053604000000067

[26] W. N. Venables and B. D. Ripley. 1999. *Modern Applied Statistics with S-PLUS.* Springer New York. https://doi.org/10.1007/978-1-4757-3121-7

[27] Michael V. Yudelson, Kenneth R. Koedinger, and Geoffrey J. Gordon. 2013. Individualized Bayesian Knowledge Tracing Models. In *Lecture Notes in Computer Science.* Springer Berlin Heidelberg, 171–180.

Using Big Data to Sharpen Design-Based Inference in A/B Tests

Adam C Sales
University of Texas at Austin
536C George I. Sánchez Building
Austin, TX 78705
asales@utexas.edu

Anthony Botelho
Worcester Polytechnic Institute 100 Institute Rd
Worcester, MA 01609
abotelho@wpi.edu

Thanaporn Patikorn
Worcester Polytechnic Institute 100 Institute Rd
Worcester, MA 01609
tpatikorn@wpi.edu

Neil T. Heffernan
Worcester Polytechnic Institute 100 Institute Rd
Worcester, MA 01609
nth@wpi.edu

ABSTRACT

Randomized A/B tests in educational software are not run in a vacuum: often, reams of historical data are available alongside the data from a randomized trial. This paper proposes a method to use this historical data–often high-dimensional and longitudinal–to improve causal estimates from A/B tests. The method proceeds in two steps: first, fit a machine learning model to the historical data predicting students' outcomes as a function of their covariates. Then, use that model to predict the outcomes of the randomized students in the A/B test. Finally, use design-based methods to estimate the treatment effect in the A/B test, using prediction errors in place of outcomes. This method retains all of the advantages of design-based inference, while, under certain conditions, yielding more precise estimators. This paper will give a theoretical condition under which the method improves statistical precision, and demonstrates it using a deep learning algorithm to help estimate effects in a set of experiments run inside ASSISTments.

1. INTRODUCTION

Randomized A/B tests hold a lot of promise for the study of student learning within intelligent tutors. Not only do they allow for causal inference without fear of confounding, but they also allow for "design-based" effect and standard error estimates that are virtually guaranteed to be unbiased [13]. These strengths are to the fact that data analysts know exactly how, and with what probability, conditions were assigned to subjects.

The traditional tools for analyzing experiments estimate effects using *only* data from the experiments themselves, discarding data from (potential) subjects that were not randomized. For instance, data from past users or from concurrent users who, for whatever reason, were not included in the A/B test, are excluded from the analysis. We refer to users such as these, with similar covariate and outcome data as the participants in the A/B test, but who were not randomized, as the "remnant." Excluding the remnant makes good statistical sense: after all, the probabilities of assignment are known only for participants in the experiment, not for the remnant. However, data from the remnant may be quite useful—in particular, the extra sample size could improve the statistical precision, i.e. reduce the standard errors, of experimental effect estimates. This is especially the case for experiments run within intelligent tutors or other big data environments. Vast amounts of log data, collected prior to the experiment, in conjunction with powerful machine-learning methods, could help sharpen causal estimates considerably.

This paper introduces a method for using the remnant in analyzing experiments, without sacrificing any of the benefits of experimentation or making additional modeling assumptions. The core of the method is residualization—predicting experimental subjects' outcomes using a model fit to the remnant, and then estimating effects using prediction residuals instead of the outcomes themselves. We call the method "remnant-based residualization," or "rebar." Rebar builds on methods suggested in [9], [7], and [1]. Rebar was first introduced in [12] as a method to reduce confounding bias in observational studies. Here we show that rebar can also reduce standard errors in randomized A/B tests, particularly in educational data mining contexts.

Most other methods incorporating machine learning into analysis of experiments, either to estimate average effects (e.g. [16]), to estimate subgroup effects (e.g. [3]), or to optimally allocate treatment (e.g. [11], [19]) use machine learning to replace, rather than complement, design based methods. This is a very promising avenue of research, but lacks the statistical guarantees of well-worn design-based estimates.

The next section will formally introduce rebar, and sections 3, 4, and 5 will illustrate it using a deep-learning model to sharpen effect estimates in a set of 22 experiments run within the ASSISTments system [14]. Section 6 will conclude.

2. REBAR

2.1 Experiments and Modeling

To learn if an intervention worked, or to figure out which of two conditions (say, condition 0 and condition 1) produces better outcomes, statistical models can often be quite helpful. To take a common example, analysts might regress an outcome Y on an indicator for condition Z, along with a vector of covariates $\boldsymbol{x}$. Then, the estimated coefficient on Z is taken as the estimated effect of condition 1 versus 0, controlling for $\boldsymbol{x}$ [18].

The shortcomings of this approach are well-known: if the vector $\boldsymbol{x}$ is missing a confounder—a covariate that predicts both subjects' choice of condition, 0 or 1, and outcomes Y—then the regression estimate will be biased. Moreover, even if there are no unmeasured confounders, if the regression model is misspecified, for instance, modeling the relationship between Y and $\boldsymbol{x}$ as linear, then the estimate will also be biased. On the other hand, a regression model may be run on all available data, producing precise (if inaccurate) estimates.

Randomized experiments correct regression's faults. If subjects are randomly assigned to conditions 0 or 1, then the difference in mean outcomes between the two groups is an unbiased estimate of the average treatment effect. More precisely, following [15] and [10], let y_{1i} be the outcome a subject i *would* experience *if* assigned to condition 1, and let y_{0i} be the outcome i would experience under condition 0. A subject's observed outcome $Y_i = y_{1i}$ if i is assigned to 1, $Z_i = 1$; $Y_i = y_{0i}$ if i is assigned to 0. (Since observed outcomes Y are a function of Z, they are random; we may model potential outcomes y_0 and y_1 as fixed.)

Under this framework, we define causal effects based on potential outcomes y_0 and y_1, rather than observed outcomes Y. An individual i's treatment effect τ_i is the difference of those two: $\tau_i \equiv y_{1i} - y_{0i}$—the difference between i's outcome under treatment versus under control. Without strong assumptions, these individual effects are not identified by the data; instead, we estimate quantities such as the average treatment effect (ATE) over all subjects $\bar{\tau} = \sum_i \tau_i/n$, or the average effect of the treatment on the treated (TOT) $\bar{\tau}_{Z=1} = \sum_i Z_i\tau_i/n_1$, where n and n_1 are the total number of subjects and the number of treated subjects, respectively. In a simple randomized experiment, the ATE and TOT have the expectation, but their estimators may have different standard errors. For the sake of simplicity, we will focus on the TOT.

Observed outcomes may be used to estimate the ATE, TOT, or other causal parameters. In particular, an unbiased estimator of the TOT is:

$$\hat{\tau} = \bar{Y}^{Z=1} - \bar{Y}^{Z=0}$$

where $\bar{Y}^{Z=1}$ is the mean of Y for treated subjects, $\sum_i Z_iY_i/n_1$, and $\bar{Y}^{Z=0}$ is the mean of Y in the control group. An unbiased estimator of the squared standard error is:

$$SE^2_{TOT} = n/(n_1n_0)s^2(\boldsymbol{Y})_{Z=0}$$

where $s^2(\boldsymbol{Y})_{Z=0}$ is the sample variance of Y in the control group. See the Technical Appendix, and [4] for more details. Estimators $\hat{\tau}$ and SE_{TOT}, and their properties, are derived solely from the experimental design, via survey sampling theory; they do not depend on the (unknown) distributions of y_1 and y_0, or any other modeling assumptions. They are "design-based."

In a randomized experiment there are no confounders. Since the probability distribution of Z is known exactly, no statistical models, or modeling assumptions, are necessary—the analysis may be "design-based" instead of model-based. In particular, the estimate $\hat{\tau}$ and its standard error derive from survey sampling theory, not the distribution of y_0 or y_1. On the other hand, any data from the "remnant" of an experiment—the set of subjects outside the experiment, who were not randomized to either condition—play no role in this analysis. Since subjects in the remnant were not randomized, there is no telling how they may differ from the $Z = 0$ or $Z = 1$ groups, in ways measured or unmeasured, and there is no telling (exactly, statistically) how their data came to be, so design-based analysis is impossible and any model fit to the remnant is most likely misspecified. However, though dropping the remnant from the analysis brings unbiasedness, it also brings a loss of precision—all that sample size, thrown away.

2.2 A Role for the Remnant

Assume the following setup: a set of users, "the experimental set" were randomized to either condition 0 or condition 1, and their outcomes $\boldsymbol{Y}$ were measured at the end of the experiment. Conditions 0 or 1 could be two different treatments, or control and treatment condition; we will refer to condition 0, as "control" and 1 as "treatment." The goal of the experiment is to estimate the TOT, $\bar{\tau}_{Z=1}$, the average effect in the treatment group. Some more subjects, the remnant, were not randomized; instead, they all received condition 0, the default (this isn't strictly necessary—the theory also works if they received condition 1, a mix of conditions, or something else altogether—but it makes things simpler). Outcomes Y were also measured for members of the remnant. Finally, a set of covariates $\boldsymbol{x}$, possibly high-dimensional, of mixed-types, and/or longitudinal, were measured for everyone, in the experimental set and in the remnant.

Experimental estimates typically drop the remnant, and pay the price of lower precision. Instead, we suggest training a machine-learning model on the remnant, and using it to "residualize" the data from the experimental set—that is, estimate effects using prediction residuals. We call this algorithm "remnant-based residualization" or "rebar." The process is as follows:

1. Using data from the remnant, train a model $\hat{y}_0(\cdot)$ to predict y_0 as a function of $\boldsymbol{x}$.

2. Validate $\hat{y}_0(\cdot)$ (using cross-validation or other techniques). if it performs well, proceed; otherwise return to step 1, choosing a different model.

3. Use $\hat{y}_0(\cdot)$ and covariates $\boldsymbol{x}$ in the experimental set to generate predicted outcomes $\hat{y}_0(\boldsymbol{x})$ and residuals, $e = Y - \hat{y}_0(\boldsymbol{x})$.

4. Estimate the TOT as a difference in mean residuals,

$$\hat{\tau}_{rebar} = \bar{e}_{Z=1} - \bar{e}_{Z=0}$$

with estimated standard error

$$SE_{rebar} = \sqrt{n/(n_1 n_0)}s(\boldsymbol{e})_{Z=0}$$

Where $s(\boldsymbol{e})_{Z=0}$ is the sample standard deviation of e in the control group.

Just like the traditional estimator $\hat{\tau}$, the rebar estimator $\hat{\tau}_{rebar}$ is design-based—its logical basis is the designed experiment, not a model. On the other hand, it harvests information from the remnant to improve upon $\hat{\tau}$.

Rebar works because the predictions $\hat{y}_0(\boldsymbol{x})$ were generated from an external sample—the remnant—and pre-treatment covariates $\boldsymbol{x}$. Subject i's prediction $\hat{y}_0(\boldsymbol{x}_i)$ will be the same whether i is assigned to 0 or to 1. Since there's no treatment effect on $\hat{y}_0(\boldsymbol{x})$, subtracting $\hat{y}_0(\boldsymbol{x})$ from Y only removes noise—not part of the treatment effect. When treatment is randomized, Z is independent of $\hat{y}_0(\boldsymbol{x})$, so, in expectation, the mean of $\hat{y}_0(\boldsymbol{x})$ will be equal across the two treatment groups. In fact, the rebar estimator can be re-written as $\hat{\tau}_{rebar} = \bar{Y}_{Z=1} - \bar{Y}_{Z=0} - (\overline{\hat{y}_0(\boldsymbol{x})}_{Z=1} - \overline{\hat{y}_0(\boldsymbol{x})}_{Z=0})$. The first term is $\hat{\tau}$, which is unbiased for the TOT. The second term is the difference in means of $\hat{y}_0(\boldsymbol{x})$, which is zero in expectation—therefore, $\hat{\tau}_{rebar}$ is unbiased. This property holds not just for the difference-in-means estimator—rebar can sharpen any treatment effect estimator that is linear in Y and unbiased.

Rebar's main tool is the model $\hat{y}_0(\cdot)$, which predicts y_0 as a function of $\boldsymbol{x}$. In EDM settings, the dimension of available covariates is often very large, and sample sizes are often large as well—machine learning algorithms make strong candidates for $\hat{y}_0(\cdot)$. $\hat{y}_0(\cdot)$ is not a statistical model *per se*, estimating the parameters of a probability distribution, but as a tool for prediction. It need not be correct in any sense, and its estimates need not be unbiased or consistent. Since $\hat{y}_0(\cdot)$ is fit on a separate sample from the experimental subjects, the process of fitting it—steps 1 and 2 above—do not affect standard errors, and model misspecification does not lead to bias.

On the other hand, for rebar to be more precise than the usual difference in means, $\hat{y}_0(\cdot)$ must be able to generate decent predictions of y_0 in the experimental set. This will be the case if $\hat{y}_0(\boldsymbol{x})$ is a good prediction of y_0—by residualizing, we subtract out the component of Y's variance that is predicted by $\hat{y}_0(\cdot)$. The variance of the rebar estimator is proportional to the difference between the mean-squared prediction error of $\hat{y}_0(\cdot)$, $MSE = ||\boldsymbol{y_0} - \hat{y}_0(\boldsymbol{x})||^2/n$ and its squared bias. (Recall that both $\hat{\tau}$ and $\hat{\tau}_{rebar}$ are unbiased estimates of the TOT; the bias here refers to $\hat{y}_0(\cdot)$'s predictions of y_0, not to treatment effect estimates.) The extent to which it outperforms the standard estimate $\hat{\tau}$, measured as percent improvement $(SE^2_{TOT} - SE^2_{rebar})/SE^2_{TOT}$, is at least as large as $\hat{y}_0(\cdot)$'s prediction R^2 in the control group, $R^2 = 1 - ||Y - \hat{y}_0(\boldsymbol{x})||^2_{Z=0}/||Y - \bar{Y}||^2_{Z=0}$ (see the Technical Appendix for derivations). If $\hat{y}_0(\cdot)$ performs poorly in the control group—so that $||Y - \hat{y}_0(\boldsymbol{x})|| > ||Y - \bar{Y}||$—then this R^2, as we have defined it, could be negative, and $\hat{\tau}_{rebar}$ will be less precise than $\hat{\tau}$; however, it will still be unbiased. The improvement $\hat{\tau}_{rebar}$ offers rests entirely on the performance of $\hat{y}_0(\cdot)$. The better we can predict how subjects would have performed in the control condition, the more precisely we can estimate treatment effects.

Since $\hat{y}_0(\cdot)$ is trained in the remnant, its performance in the experimental set (as measured by, e.g. prediction R^2) will be hard to gauge at the outset. If the distribution of Y, conditional on $\boldsymbol{x}$, differs widely from the between the two sets, $\hat{y}_0(\cdot)$'s performance may suffer in extrapolation. This problem is not fatal: the rebar estimate is unbiased regardless of $\hat{y}_0(\cdot)$'s properties. However, a model with poor predictive power in the experimental set will not reduce standard errors substantially, and may increase them. Of course an analyst may calculate $\hat{y}_0(\cdot)$'s R^2 in the experimental set, but choosing a model based on Y induces dependence between Y and $\hat{y}_0(\boldsymbol{x})$, and may cause bias. Models trained on a subset of the remnant that resembles the experimental set—or which weight such a remnant more heavily—may perform better than those trained on the entire remnant.

The previous discussion assumed simple randomization. However, rebar easily extends to more complex designs, including experiments with more than two treatment conditions. Further, as we will illustrate below, rebar can be extended to regression estimators of causal effects as well, modeling low-dimensional covariates within sample and high-dimensional covariates out of sample.

3. DATA: 22 EXPERIMENTS AND MORE

The 22 experiment dataset is a feature-rich dataset on students who participated in randomized controlled trials (RCTs) ran inside a free, online tutoring called ASSISTments [14]. This dataset consists of student-level data from 8,205 unique students participating in 22 A/B tests, 14,947 unique student-RCT pairs in total.

These RCTs were run within skill builders. Inside ASSISTments, a skill builder is a type of problem set that requires students to practice solving problems until they master the associated skill. Skill mastery is determined by the student's ability to answer a certain number of problems correctly, usually three, in a row.

This feature-rich dataset includes 30 features, including both categorical features, such as student grade levels, and numerical features, such as student performances prior to the experiment. This dataset also includes two dependent measures. The first dependent measure, "completion," is whether the student completed the assignment and achieved mastery. The other dependent measure is the number of problems attempted; for students who achieved mastery, this may be interpreted as mastery speed. The analysis in this paper will focus on the first dependent measure, completion.

4. DEEP LEARNING TO PREDICT COMPLETION

As described in Section 2.1, the model $\hat{y}_0(\cdot)$ is an integral part of the rebar methodology with the purpose of producing predicted outcome y_0 as a function of $\boldsymbol{x}$. The methodology does not rely on a specific type of model, nor any specific algorithm to be used so long as an estimate for the outcome variable of interest is generated by the model from included covariates. As also stated in that section, the accuracy of the model, however poor, does not lead to bias. That said, models that are more accurate at estimating the outcome variable of interest will likely lead to better estimates of treatment effects. Deep learning models have been previously applied in educational contexts with promising results, often reporting higher performance over existing methods [8][6][2]. While the application of such methods is not appropriate to all problem applications due to the size and complexity, the use of such models in this work is justified due to 1) the scale of data available for model training and 2) the inconsequence of producing an uninterpretable model (e.g. the significance and coefficients of individual variables in the model are not intended for study or knowledge discovery). What is needed, again, is simply a prediction model.

We develop and apply a type of deep learning model known as a long short term memory (LSTM) [5] network. This model is a variant of a recurrent neural network (RNN) [17] that is commonly applied to time series data to model temporal relationships within the sequences. The model produces its estimates for each time step by utilizing both covariates provided corresponding to the current time step as well as information from all previous time steps within the series. As such, the model is developed as a sequence-to-sequence method that observes a sequence of student data as input and produces a sequence of outcome estimates of equal length. The model structure utilizes a 3-layer design, with an input layer feeding into a recurrent hidden layer (represented as a layer also connected to itself in previous time steps), before then proceeding to an output layer.

Two separate datasets are used to train and apply the model. The application dataset, comprised of student data from the 22 experiment dataset combined with assignment-level information for all work each student started before beginning the respective experiment. In an attempt to reduce the complexity of the data from which the model must learn, the sequence length of student assignment history is limited such that no more than 10 prior assignments are included for each student. In other words, students who were in a single experiment have a sequence length of 10, with the last time step representing the most recent assignment prior to beginning the experiment. Conversely, students in multiple students may exhibit sequences longer than 10 if participation in the experiments was separated by fewer than 10 assignments. The dataset is comprised of data from 8,297 distinct students and a total of 130,935 student assignments.

The second dataset, used to train and validate the model, is comprised of student data exclusive to that comprising the 22 experiment dataset. Student data, again non-inclusive of students within the 22 experiment dataset, is collected from the non-experimental problem sets found in the application dataset. From these, assignments are randomly sampled, with which the dataset is constructed using the 10 most recent assignments before students begin the sampled assignment. This step, helps to ensure a similar structure of the dataset to that of the application set. Again, for purposes of validity, it is important to stress that no students are found in both the training and application datasets. The dataset contains data from 134,141 distinct students and a total of 686,590 student assignments.

The model uses just 4 assignment-level covariates per time step to predict assignment-level performance on the subsequent assignment. These covariates include the simple measures of completion of the assignment, the number of problems attempted, the number of problems completed, and a measure of inverse mastery speed; this last measure is a transformation of mastery speed, using 1 divided by the number of problems when the assignment was complete, or 0 when the assignment was not completed. While simple in the number of covariates, again, the model also uses information from previous time steps, adding to its complexity (i.e. time step 2 is informed by time step 1, time step 3 is informed by time steps 2 and 1, etc.). The model produces two values per time step corresponding with the desired outcome variable of completion of the next assignment, and also an estimate of inverse mastery speed on the next assignment, using a combined cost of these two measures to update model parameters during training; this second measure was included as it is believed the model may better learn from the data by observing a continuous variable in addition to the binary value of completion and also acts as an example as to how future works may utilize the same methodology to observe beyond the measure of completion presented in this work.

The model is first evaluated using a 5-fold student-level cross-validation. The model is trained for multiple epochs, or training cycles through the data, using a 30% holdout set, sampled from the training set of each fold, to determine the stopping point of model training; this holdout set also helps stop the model training process before overfitting is detected. It is found that the model produces average AUC of 0.81 and an RMSE of 0.34 for next assignment completion over the 5 folds. Once completed, a final model is trained over the entirety of the training dataset and applied to the application dataset, which has acted as a holdout set during the training and validation process. The next assignment completion estimates, collected from the most recent assignment before students begin each experiment, is then used as the estimated value of completion that is used in subsequent steps of the rebar analyses.

5. RESULTS

We estimated treatment effects of interventions on skill-builder completion for the 22 experiments using both raw outcomes $\boldsymbol{Y}$, the usual approach, and using $\boldsymbol{e} = \boldsymbol{Y} - \hat{\boldsymbol{y}}_{\boldsymbol{0}}(\boldsymbol{x})$, the rebar estimator. We also estimated standard errors in both cases. We used difference in means estimators, so the effect estimates are in units of percentage points—how much more likely were students to finish skill builders under the treatment condition than under control.

Figure 1 shows improvement in precision of the rebar estimator over the usual estimator: the difference of the two standard errors, divided by the usual standard error, $(SE_{usual} - SE_{rebar})/SE_{usual}$. The x-axis shows the predic-

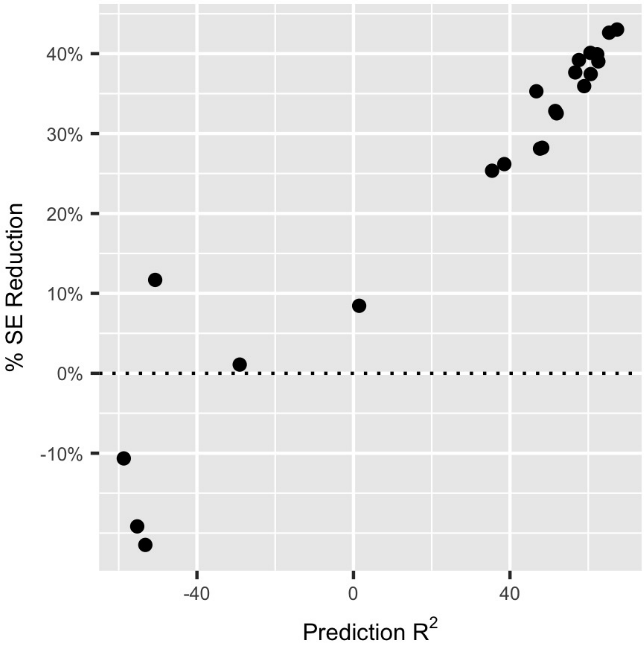


Figure 1: The improvement in precision of effect estimates as a percentage of the usual precision estimate, $[SE(\hat{\tau})-SE(\hat{\tau}_{rebar})]/SE(\hat{\tau})$, plotted as a function of $\hat{y}_0(\cdot)$'s prediction R^2 in the experimental set.

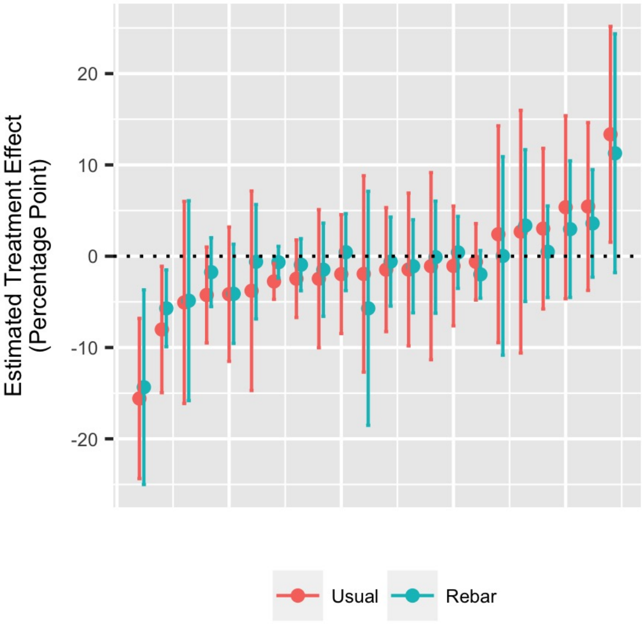


Figure 2: Effect estimates and 95% confidence intervals for the 22 experiments, using both the usual and rebar estimates. Experiments are ordered by their estimated effect.

tion R^2 of the deep learning model when extrapolated to each of the 22 experimental datasets. In 19 of the datasets, the rebar standard errors were lower than the usual standard errors. In 15 of those datasets, there was a greater than 25% improvement, and in four datasets the improvement was greater than 40%. The extent of the reduction in standard error corresponded closely to the prediction R^2 of $\hat{y}_0(\cdot)$, with the most dramatic improvements occurring when $R^2 \gtrsim 0.5$.

Figure 2 shows the estimated treatment effects and approximate 95% confidence intervals (two standard errors in each direction) for the two sets of estimators. In all but three cases, the rebar estimate was slightly closer to zero than the usual estimate. This is what we would expect if most of the true effects were null, so that reducing the noise of the treatment effect estimates would draw them closer to their true values. For that reason, although rebar reduced the standard errors in almost all of the experiments, it did not cause any of the non-significant results to become statistically significant. In fact, in two cases it had the opposite effect; though this may be disappointing for researchers, it is probably more accurate.

We also used linear regression to estimate treatment effects, regressing either indicators for completion or prediction errors on indicators for treatment assignment and two covariates: the proportions of students' prior skill builders completed and the proportions of prior skill builder problems students worked that they answered correctly. The results, available upon request, are nearly identical. Although the two covariates improved precision slightly, rebar continued to dominate the usual estimate.

6. DISCUSSION

The rich, high-dimensional, fine-grained data that educational technology makes available should be a boon to causal inference. However, big data is subject to the same maladies as small data—confounding from unmeasured variables, and model misspecification. Classical randomized experiments remain relevant.

The same may not be true for classical design based estimators. Big data may not be able to correct unmeasured confounding and may exacerbate model misspecification, but we have shown that it can play a significant role reducing the standard errors of treatment effect estimates. The method we proposed here retains all of the statistical properties that recommend design-based estimators, while, in most cases, delivering substantially lower standard errors. We demonstrated the method's effectiveness using a cutting-edge deep learning algorithm trained to log data from ASSISTments which yielded impressive gains in precision when used to analyze a set of 22 experiments.

Rebar's most important tool in this exercise was the deep learning algorithm, which in 17 of the 22 experiments predicted completion better than the within-sample proportion. In general, designing prediction algorithms that perform well in the target dataset is the central challenge to effectively implementing rebar. Along the same lines, the most important open question is how to design diagnostics for prediction performance that do not rely on "peeking" at the experimental outcomes. One such diagnostic, termed "proximal validation," was described in [12]—extending it to experimental studies and showing that it works is the next step in developing this method.

Wedding classical randomization-based causal inference with modern machine learning and big data can yield unbiased, robust, precise treatment effect estimates in technology-based educational datasets.

Technical Appendix

This discussion roughly follows [4], Section 1.1. The TOT $\sum_i Z_i(y_{1i} - y_{0i})/n_1$ may be re-written as $n_1^{-1}\left(\sum_i Y_i - \sum_i y_{0i}\right)$, the difference between the total of Y across both treatment groups, and the total that would have been observed had everyone received the control condition. The first sum is known exactly, but the second must be estimated using data from the control group. From elementary survey sampling, $n\bar{Y}^{Z=0}$ is unbiased for $\sum_i y_{0i}$. Further, the standard deviation of $n\bar{Y}^{Z=0}$ is $\sqrt{n^2(1-n_0/n)s^2(\boldsymbol{y_0})/n_0} = \sqrt{n/(n_1 n_0)}s^2(\boldsymbol{y_0})$, where $s^2(\boldsymbol{y_0})$ is the sample variance, over the whole sample, of y_0, and $1 - n_0/n = n_1/n$ is the finite population correction. Finally, due to random sampling, $s^2(\boldsymbol{Y})_{Z=0}$ is an unbiased estimator for $s^2(\boldsymbol{y_0})$. Substituting $\sigma_{Z=0}$ for σ_0 and dividing by n_1 gives the expression for SE_{TOT}.

Each individual treatment effect τ_i is the same whether the outcome (dependent variable) is Y or e:

$$e_1 - e_0 = (y_1 - \hat{y}_0(\boldsymbol{x})) - (y_1 - \hat{y}_0(\boldsymbol{x})) = y_1 - y_0$$

since $\hat{y}_0(\boldsymbol{x})$ is invariant to treatment. Therefore, the theory supporting standard estimates $\hat{\tau}$ and SE_{TOT} applies equally to $\hat{\tau}_{rebar}$ and SE_{rebar}. In particular, $\hat{\tau}_{rebar}$ is unbiased for the TOT with consistent standard error estimate SE_{rebar}, due to survey sampling theory.

The sample variance of e in the control group is

$$\begin{aligned}
s^2(e)_{Z=0} &= \frac{||\boldsymbol{e} - \bar{e}||^2_{Z=0}}{n_0 - 1} \\
&= \frac{||\boldsymbol{Y} - \boldsymbol{\hat{y}_0(x)} - (\bar{Y} - \overline{\hat{y}_0(\boldsymbol{x})})||^2_{Z=0}}{n_0 - 1} \\
&= \frac{||\boldsymbol{Y} - \boldsymbol{\hat{y}_0(x)}||^2_{Z=0}}{n_0 - 1} - \frac{n_0}{n_0 - 1}\left(\bar{Y}_{Z=0} - \overline{\hat{y}_0(\boldsymbol{x})}_{Z=0}\right)^2
\end{aligned}$$

or the MSE of $\hat{y}_0(\cdot)$ in the control group, minus its squared bias. Since the squared bias is always positive,

$$s^2(e)_{Z=0} \le \frac{||\boldsymbol{Y} - \boldsymbol{\hat{y}_0(x)}||^2_{Z=0}}{n_0 - 1}$$

Therefore, the ratio of the estimated rebar standard error to the usual TOT standard error is:

$$\left(\frac{SE_{rebar}}{SE_{TOT}}\right)^2 = \frac{s^2(e)_{Z=0}}{s^2(Y)_{Z=0}} \le \frac{||\boldsymbol{Y} - \boldsymbol{\hat{y}_0(x)}||^2_{Z=0}}{||\boldsymbol{Y} - \bar{Y}||^2_{Z=0}} = 1 - R^2_{Z=0}$$

with equality if $\hat{y}_0(\cdot)$ is unbiased.

7. ACKNOWLEDGEMENTS

Ben B Hansen provided foundational ideas and valuable guidance in the early stages of this project. This work was partially funded by: NSF (IIS-1636782, ACI-1440753, DRL-1252297, DRL-1109483, DRL-1316736, DGE-1535428 & DRL-1031398), the US Department of Education (IES R305A120125 & R305C100024 and GAANN), and the ONR.

8. REFERENCES

[1] P. M. Aronow and J. A. Middleton. A class of unbiased estimators of the average treatment effect in randomized experiments. *Journal of Causal Inference*, 1(1):135–154, 2013.

[2] A. F. Botelho, R. S. Baker, and N. T. Heffernan. Improving sensor-free affect detection using deep learning. In *International Conference on Artificial Intelligence in Education*, pages 40–51. Springer, 2017.

[3] W. Duivesteijn, T. Farzami, T. Putman, E. Peer, H. J. Weerts, J. N. Adegeest, G. Foks, and M. Pechenizkiy. Have it both waysâĂŤfrom a/b testing to a&b testing with exceptional model mining. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 114–126. Springer, 2017.

[4] B. B. Hansen and J. Bowers. Attributing effects to a cluster-randomized get-out-the-vote campaign. *Journal of the American Statistical Association*, 104(487):873–885, 2009.

[5] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[6] M. Khajah, R. V. Lindsey, and M. C. Mozer. How deep is knowledge tracing? *arXiv preprint arXiv:1604.02416*, 2016.

[7] T. Patikorn, D. Selent, N. T. Heffernan, J. E. Beck, and J. Zou. Using a single model trained across multiple experiments to improve the detection of treatment effects. In *Proceedings of the 10th International Conference of Educational Data Mining*, 2017.

[8] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, pages 505–513, 2015.

[9] P. R. Rosenbaum et al. Covariance adjustment in randomized experiments and observational studies. *Statistical Science*, 17(3):286–327, 2002.

[10] D. B. Rubin. Bayesian inference for causal effects: The role of randomization. *The Annals of statistics*, 6(1):34–58, 1978.

[11] P. Rzepakowski and S. Jaroszewicz. Decision trees for uplift modeling with single and multiple treatments. *Knowledge and Information Systems*, 32(2):303–327, 2012.

[12] A. C. Sales, B. B. Hansen, and B. Rowan. Rebar: Reinforcing a matching estimator with predictions from high-dimensional covariates. *Journal of Educational and Behavioral Statistics*, 43(1):3–31, 2018.

[13] P. Z. Schochet. Statistical theory for the" rct-yes" software: Design-based causal inference for rcts. ncee 2015-4011. *National Center for Education Evaluation and Regional Assistance*, 2015.

[14] D. Selent, T. Patikorn, and N. Heffernan. Assistments dataset from multiple randomized controlled experiments. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pages 181–184. ACM, 2016.

[15] J. Splawa-Neyman. On the application of probability theory to agricultural experiments. essay on principles. section 9. *Roczniki Nauk Rolniczych*, 10:1–51, 1923. Translated and edited by D. M. Dabrowska and T. P. Speed for *Statistical Science*, 5(4):456–72, 1990.

[16] M. J. Van der Laan and S. Rose. *Targeted learning: causal inference for observational and experimental data*. Springer Science & Business Media, 2011.

[17] R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[18] G. U. Yule. An investigation into the causes of changes in pauperism in england, chiefly during the last two intercensal decades (part i.). *Journal of the Royal Statistical Society*, 62(2):249–295, 1899.

[19] S. Zhao and N. Heffernan. Estimating individual treatment effect from educational studies with residual counterfactual networks. In *Proceedings of the 10th International Conference on Educational Data Mining, EDM*, 2017.

Exploring Potential Effectiveness of Jaccard index to Measure Treatment Integrity in Virtual Reality-based Social Training Program for Children with High-functioning Autism

Jewooong Moon
Educational Psychology and Learning Systems
Florida State University
600 W College Ave, Tallahassee
jewoongmoon87@gmail.com

Fengfeng Ke
Educational Psychology and Learning Systems
Florida State University
600 W College Ave, Tallahassee
fke@fsu.edu

ABSTRACT

With the advent of virtual reality-based social training for autistic children, treatment integrity of training has been a key measure to determine the quality of the intervention. This study introduces Jaccard index to evaluate autistic children's behaviors in guided discovery-designed virtual training. The finding from this study confirmed that Jaccard index could potentially gauge how much autistic children behave corresponding to required social behaviors in virtual reality-based training.

Keywords

Virtual reality, Jaccard Index, Social skill training, and Intervention integrity and quality, and High functioning autism.

1. INTRODUCTION

Virtual reality-based social training has been one of promising approaches to facilitate socially-required interaction skills of children with high-functioning autism (HFA) (Didehbani, Allen, Kandalaft, Krawczyk, & Chapman, 2016; Stichter, Laffey, Galyen, & Herzog, 2014). According to weak central coherence (WCC) theory, children with HFA usually have normal cognitive abilities to process their information but low capabilities to figure out malleable social senses to identify "big picture" of the context (Mottron, Burack, Iarocci, Belleville, & Enns, 2003). The virtual reality-based social training may reduce autistic children's social anxiety level as well as provoke their motivation when employing social interaction in the training (Stichter et al., 2014). This could allow HFA children to smoothly become familiar with social interaction in real world. In virtual reality-based training, it is normally designed to promote learners' participation together with much social interaction with others in guided discovery environment (Didehbani et al., 2016). As guided discovery learning design for social training for children with HFA, the designers are supposed to purposefully plan interactive events to initiate their social skills. The flow of the virtual training session is intended to provoke HFA students' socially-acceptable morale behaviors. It means that the training by the guided discovery may allow children with HFA themselves to follow socially-required behaviors to provoke ideal social skills.

Currently, the key issue as to virtual reality-based training should be assessment. Much research as to social training for children with HFA (Didehbani et al., 2016; Stichter et al., 2014) has addressed how to maintain reliable effectiveness of the training program itself. Treatment integrity is generally a crucial degree to determine how much training for autistic children could be reliable to be implemented as evidence-based practices (Ke, Whalon, & Yun, 2017). Even if a quality indicator of a training intervention for children with HFA has been emphasized dominantly, there is still few approaches to gauge how children with HFA coherently estimate what they are supposed to behave in their training.

Derived by the ideas from serious game analytics (Loh & Sheng, 2015), utilizing Jaccard index could be an alternative technique to capture sequential steps whether children with HFA clearly replicate standard behaviors as what each training session deliberately facilitates. Jaccard index is one of measures to evaluate users' behaviors compared with experts' behaviors as standard one. In other words, it quantifies the ratios of behavior sequences of one user compared to standard behavioral sequences. The kernel of Jaccard index should be the comparison between behaviors of children with HFA and normally acceptable social behaviors in each session for social training. The research question of this study is following: How does Jaccard index of each child with HFA reflect the improvement of social behaviors in virtual reality-based social training?

2. METHOD

The samples of this study are nine children with HFA (Male = 8 / Female = 1), who attended virtual reality-based social training sessions. The operating system to simulate virtual reality was Opensimulator, which simulates collaborative virtual environment. With two facilitators for training in virtual reality, all children with HFA in this study went over three guided discovery-oriented training sessions: (1) roleplaying as a server in a restaurant, (2) scavenge hunting, and (3) librarian interviews. All sessions of each participant in this study were video-captured. On average, each session lasted around 60 minutes. Via using content analyses of captured videos of the participants as well as preplanned design documents of each session, two evaluators created the rubric of standard behaviors in each session. The rubric has been iteratively revised until two evaluators acquire 100 % agreement of the rubric. Derived from the concept of Jaccard index, the rubric described each different number of critical behaviors in each session respectively. Based on the rubric, the study calculated each individual's Jaccard index scores ($0 < X_{Jaccard} < 1$).

$$\text{Jaccard Index score of a participant in each session} = \frac{(\text{A participant's behavior}) \cap (\text{Standard behaviors in a rubric})}{(\text{A participant's behavior}) \cup (\text{Standard behaviors in a rubric})}$$

Figure 1. Equation to calculate Jaccard index score of each participant in each session.

Figure 1 is the equation for calculating Jaccard index scores to estimate their behaviors in each session. To confirm the reliability of the index, additional in-depth video analyses were also implemented.

3. Findings

Figure 2 shows the average scores of Jaccard index of each child with HFA as well as the scores across participants and types of training sessions. Most participants in the study had higher Jaccard index scores in the roleplaying session (M=.87, SD= .20) compared to those in other sessions (Scavenger hunting = .51 / Librarian interview = .37). As shown by using in-depth video analysis, the participants, who had scores lower than .5, were confirmed that they were mostly inattentive toward the activity or did not figure out how to start with their actions to resolve a given task in the sessions.

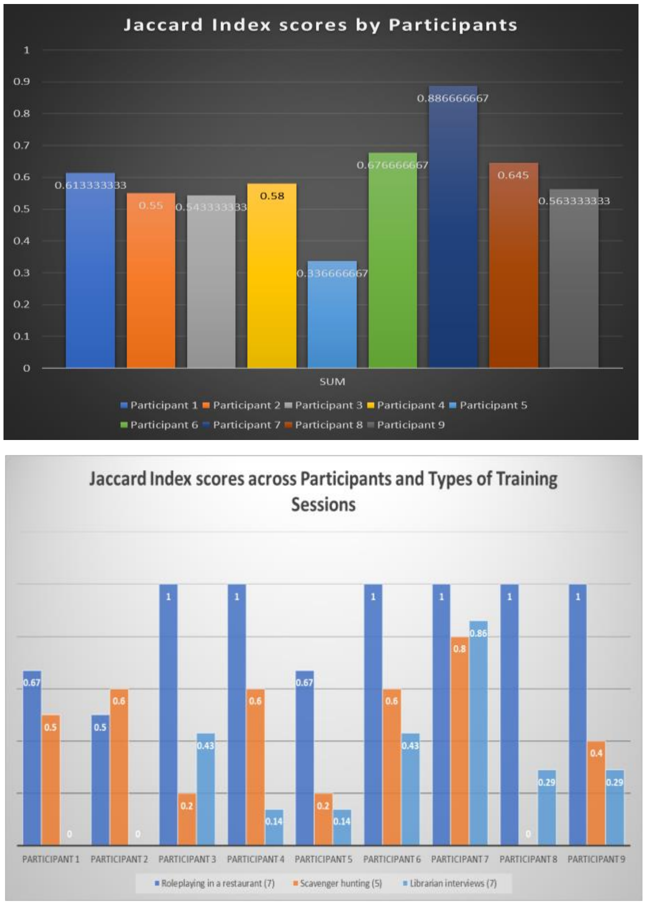


Figure 2. Jaccard index scores across each participant as well as types of training sessions

4. CONCLUSION

As findings of this study shown by Jaccard index, most participants might have difficulties to follow proposed steps of behaviors in librarian interviews while they were usually able to act as a server of the restaurant in roleplaying. Via using Jaccard index, this study potentially proposed that additional scaffolding for children with HFA should be necessary for librarian interviews, which were likely to make learners be much responsible for interacting with an interviewee as face-to-face interaction compared to other sessions.

Treatment integrity in social training for autistic children could be a determinant whether an intervention deliberately provokes proposed social behaviors. In a same vein with this context, this study revealed that Jaccard index could be promising to represent whether children with HFA clearly behave by the intervention in order to ensure the quality of treatment integrity.

5. ACKNOWLEDGMENTS

This work was supported by the Spencer Foundation on 2014.

6. REFERENCES

[1] Didehbani, N., Allen, T., Kandalaft, M., Krawczyk, D., & Chapman, S. 2016. Virtual reality social cognition training for children with high functioning autism. *Computers in Human Behavior, 62*, 701-711. DOI=https://doi.org/10.1016/j.chb.2016.04.033.

[2] Ke, F., Whalon, K., & Yun, J. 2017. Social skill interventions for youth and adults with autism spectrum disorder: A systematic review. *Review of Educational Research*, 62. DOI= doi:10.3102/0034654317740334

[3] Loh, C. S., & Sheng, Y. 2015. Measuring the (dis-)similarity between expert and novice behaviors as serious games analytics. *Education and Information Technologies*, 20(1). 5-19, DOI= doi:10.1007/s10639-013-9263-y

[4] Mottron, L., Burack, J. A., Iarocci, G., Belleville, S., & Enns, J. T. 2003. Locally oriented perception with intact global processing among adolescents with high-functioning autism: evidence from multiple paradigms. *Journal of Child Psychology and Psychiatry*, 44(6). 904-913, DOI= doi:10.1111/1469-7610.00174

[5] Stichter, J. P., Laffey, J., Galyen, K., & Herzog, M. 2014. iSocial: Delivering the social competence intervention for adolescents (SCI-A) in a 3D virtual learning dnvironment for youth with high functioning autism. *Journal of Autism and Developmental Disorders*, 44(2). 417-430, DOI= doi:10.1007/s10803-013-1881-0

A Tool for Preprocessing Moodle data sets

Javier López-Zambrano[1], José Antonio Martinez[2], Jesús Rojas[2], Cristóbal Romero[2]
[1]ESPAM MFL, Faculty of Computation, 131106, Calceta, Ecuador
[2]University of Cordoba, Dept. of Computer Science, 14071, Córdoba, Spain
jlopez@espam.edu.ec, i32marej@uco.es, i32roraj@uco.es, cromero@uco.es

ABSTRACT

This paper describes a desktop Java tool for allowing instructors to preprocess Moodle data sets. Our idea is to provide instructors with an easy to use tool for preparing the raw Excel students data files directly downloaded from Moodle's courses interface. Several traditional preprocessing techniques are considered to transform input data into well-formatted data sets that can be later used by most of the popular data mining frameworks.

Keywords

Moodle's students data, data preprocessing, data mining tool.

1. INTRODUCTION

Nowadays, there is a great interest in analyzing and mining any students' usage/interaction information gathered by Learning Management Systems (LMS) such as Moodle [1]. However, to obtain and preprocess these data can be an arduous and tedious task [2]. Generally, it is necessary to know SQL language as well as to be an user with administrator role in order to have access to all the course information. And to our knowledge there isn't any specific Moodle data mining tool for preprocessing [2]. So, in order to resolve these problems, we have developed an easy to use Java GUI application oriented to be used by non-expert users in data mining and SQL, such as instructors. Our idea is to provide the instructor of a Moodle course the possibility of using Excel files directly downloaded from Moodle's interface without a labor and time-intensive preprocessing step. Finally, the obtained files from our desktop tool are well-formatted datasets that can be used by most of the well-known data mining frameworks (Weka, RapidMiner, Knime, R, etc.) for applying data mining algorithms.

2. TOOL DESCRIPTION

Our Moodle data preprocessing desktop tool has been developed in Java language and it includes six main steps and taps (see Figure 1).

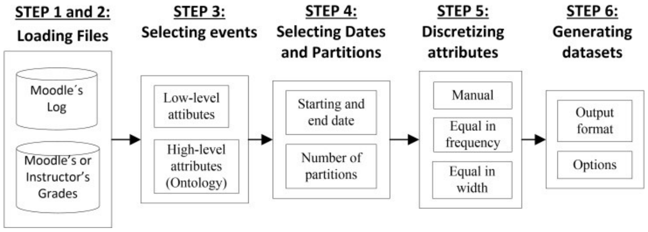
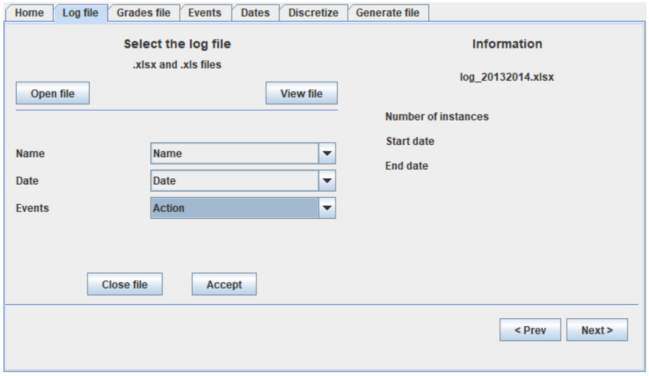


Figure 1: Preprocessing flow

2.1 Log file selection

This tab enables a log file (directly downloaded from Moodle's course interface in spreadsheet Excel format) to be opened/loaded. After that, it shows the content of the file and allows selecting the specific columns where the required information is located (Name of the students, Date and Events). This tab also provides basic information about the loaded file such as the total number of records, and the first and last update for all the records (see Figure 2).



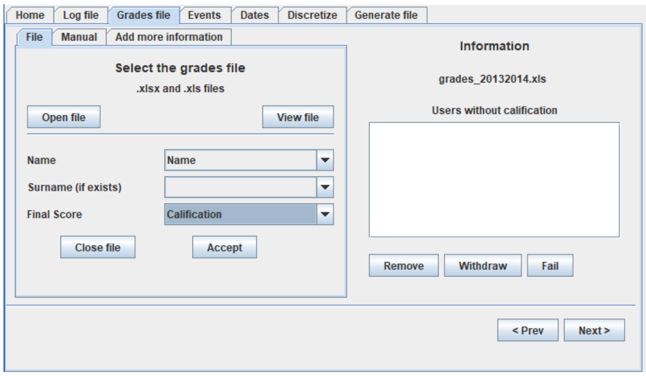
Information
log_20132014.xlsx
Number of instances
Start date
End date

Name: Name
Date: Date
Events: Action

Figure 2: Selecting a log file.

2.2 Grades file selection

This tab is used by instructors to load a file (in spreadsheet Excel format) containing the students' grades (directly downloaded from Moodle or provided by the own instructors). Instructors can also fill in the students' mark manually (see Figure 3). Finally, those students with no final mark in the course can be removed, set as fail or even set as withdraw.



Information
grades_20132014.xls
Users without calification

Name: Name
Surname (if exists):
Final Score: Calification

Figure 3: Loading a grades file.

2.3 Events selection

This tab allows the instructor to select what events (all of them or just a few) should be used as attributes in the final dataset. It is also possible to group these raw events in new high level attributes manually or automatically by using an ontology (see Figure 4). This ontology can be created, saved, loaded and viewed.

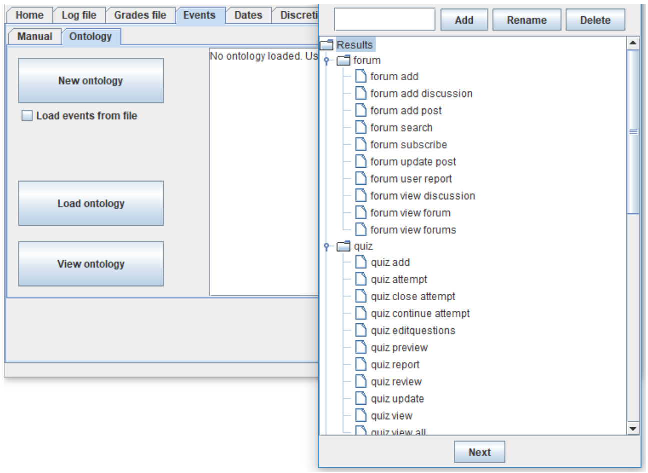
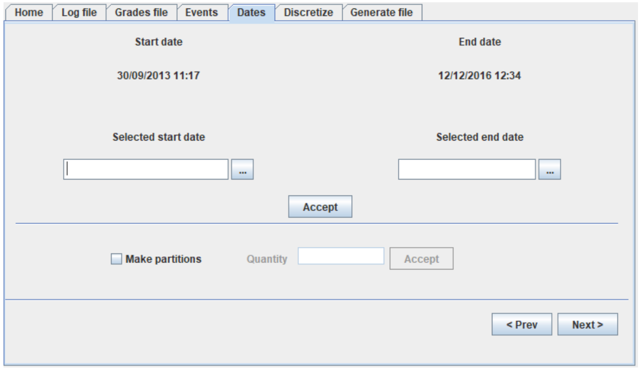


Figure 4: Selecting events using an ontology.

2.4 Date and partitions selection

The specific starting and ending date of the course can be established from this tab in order to use only the events that occurred between these dates (see Figure 5). It is also possible to specify whether the user requires a single summarization file or a number of cumulative data partitions (e.g. one per week/month).



Start date: 30/09/2013 11:17

End date: 12/12/2016 12:34

Figure 5: Selecting dates and partitions.

2.5 Discretization

For the sake of transforming those attributes or variables defined in a continuous domain/range into discrete values, this tab provides the option of performing a manual discretization as well as traditional techniques such as equal-width or equal-frequency (see Figure 6).

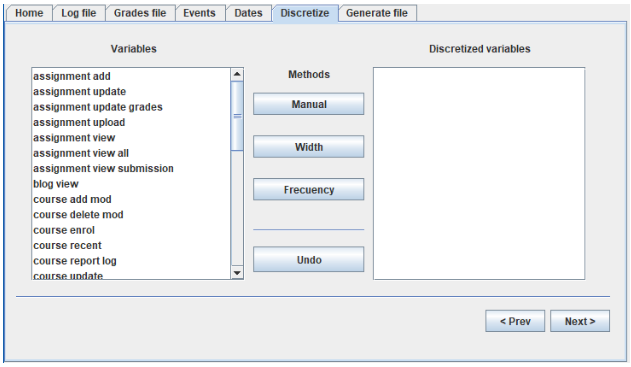
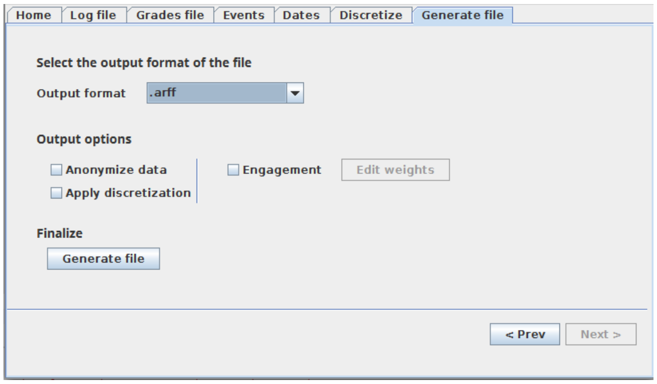


Figure 6: Discretizing variables.

2.6 Dataset generation

Finally, this last tab allows the instructor to generate the preprocessed data file, or several data files in case he/she selected several partitions that can be downloaded in three different file formats: .ARFF (Attribute-Relation File Format), .CSV (Comma-Separated Values) and .XLS (eXceL Spreadsheet). This tab includes additional options such as data anonymization and previous discretization techniques (see Figure 7). It also gives the possibility to generate a student's engagement variable that unifies the time, in minutes and days that each student has been connected in Moodle, as well as the total number of records/instances of each student in the log file.



Output format: .arff

Figure 7: Generating preprocessed datasets.

3. ACKNOWLEDGMENTS

The authors acknowledge the financial subsidy provided by Spanish Ministry of Science and Technology TIN2017-83445-P.

4. REFERENCES

[1] Luna, J. M., Castro, C., Romero, C. 2017. MDM tool: A data mining framework integrated into Moodle. Computer Applications in Engineering Education, 25(1), 90-102

[2] Romero, C., Romero, J.R., Ventura, S. 2014. A Survey on Pre-processing Educational Data. Educational Data Mining: Applications and Trends, Springer Series Studies in Computational Intelligence. 29-64, 2014.

Exploring Learning-Facilitating Game Actions via Sequential Analysis

Fengfeng Ke & Jewoong Moon
Florida State University
Educational Psychology and Learning
Systems
Tallahassee, FL 32306-4453
8506448794, +1
fke@fsu.edu, jmoon3@fsu.edu

ABSTRACT

This study explored game-based learning behaviors via sequential data mining. A systematic behavior analysis was conducted with a total of 98 videos from 25 middle school students who played *E-Rebuild* – an architectural simulation game based mathematical learning platform. A transition matrix analysis in sequential data mining was implemented to explore the game actions that promoted cognitive and content engagement with high probability.

Keywords

Sequential analysis, game-based learning, behavior analysis.

1. INTRODUCTION

Prior research indicated that game-based learning engagement is not a naturalistic occurrence in educational gaming and should be purposefully designed and supported (Ke, Xie, & Xie, 2015). Learners usually interact with multiple game objects and events in their play. However, it is challenging to predict and explain what game actions are most likely to promote learning engagement. In this study, we explored learners' in-game behaviors to *predict the association between diverse game actions and their learning engagement via sequential data mining*.

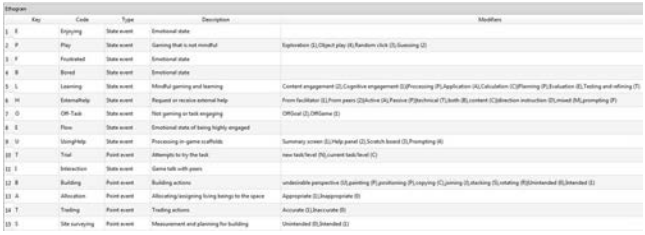
2. METHOD

2.1 Game-based learning platform

The game *E-Rebuild* is a single-player, multi-episode 3D architecture game of which the overall game goal is to rebuild a natural disaster damaged space to fulfill diverse design parameters or needs. The targeted math content topics by E-Rebuild are (a) ratio and proportional relationships, and (b) angle measure, area, and surface area. A learner in E-Rebuild will perform the following core game actions: *surveying* the construction site, *collecting* construction items, a series of *building* actions (including object positioning and stacking), materials *trading*, and space and resource *allocation*.

2.2 Participants and data collection

A total of 98 videos capturing 25 (male=15, female=10) middle school students' game-play behaviors and facial expressions were analyzed. Each video lasted around 50 minutes. Multiple coders conducted an exploratory behavior analysis to construct a gaming behavior coding framework (Figure 1). The framework was refined until the coders achieved a 100% inter-rater agreement on the categories, definitions, and exemplifying events of core behaviors. As Figure 1 shows, action categories represent the gameplay actions in which the learners engaged. The outcome variable, learning engagement, comprised two types of subordinate states: generic cognitive engagement (e.g., planning, evaluation, refining) and content engagement (e.g., information processing, knowledge application, calculation). Using the coding framework and BORIS (Friard & Gamba, 2016), four trained coders coded 20% of gaming recordings blindly, with the interrater agreement being .82. After further peer debriefing to resolve the disagreements, the coders then coded an equal part of the remained recordings independently.



Ethogram

	Key	Code	Type	Description	Modifiers
1	E	Enjoying	State event	Emotional state	
2	P	Play	State event	Gaming that is not mindful	Exploration (1),Object play (4),Random click (3),Guessing (2)
3	F	Frustrated	State event	Emotional state	
4	B	Bored	State event	Emotional state	
5	L	Learning	State event	Mindful gaming and learning	Content engagement (2),Cognitive engagement (1)|Processing (P),Application (A),Calculation (C)|Planning (P),Evaluation (E),Testing and refining (T)
6	H	ExternalHelp	State event	Request or receive external help	From facilitator (1),From peers (2)|Active (A),Passive (P)|technical (T),both (B),content (C)|direction instruction (D),mixed (M),prompting (P)
7	O	Off-Task	State event	Not gaming or task engaging	OffGoal (2),OffGame (1)
8	E	Flow	State event	Emotional state of being highly engaged	
9	U	UsingHelp	State event	Processing in-game scaffolds	Summary screen (1),Help panel (2),Scratch board (3),Prompting (4)
10	T	Trial	Point event	Attempts to try the task	new task/level (N),current task/level (C)
11	I	Interaction	State event	Game talk with peers	
12	B	Building	Point event	Building actions	undesirable perspective (U),painting (P),positioning (P),copying (C),joining (J),stacking (S),rotating (R)|Unintended (0),Intended (1)
13	A	Allocation	Point event	Allocating/assigning living beings to the space	Appropriate (1),Inappropriate (0)
14	T	Trading	Point event	Trading actions	Accurate (1),Inaccurate (0)
15	S	Site surveying	Point event	Measurement and planning for building	Unintended (0),Intended (1)

Figure 1. Coding framework via BORIS

2.3 Sequential data mining

With the behavior coding results, we then employed a transition matrix analysis in sequential data mining (Friard & Gamba, 2016) to explore game actions that predict learning engagement with certain probabilities. The underlying principle from the transition matrix analysis is *hidden Markov chain* algorithm (Martin & Sherin, 2013). In comparison with other algorithms in sequential analyses, the *hidden Markov chain* can better illustrate a causal relationship between event variables and predictors (Hou, 2015). Different from a *lag-sequential analysis* (Gottman & Roy, 1990), in this study we focused on the exploration of potentially-prominent game actions that promote learning engagement but not to compare whether a predictive model is statistically significant.

In hidden Markov chain, there are the *transition probability* representing probability values between two events in a same behavior category and the *emission probability* representing the probability that one event can predict another event in each different category. In this study we examined mainly the *emission probability* to identify the game events or actions that predict two types of learning engagement (Hand, Mannila, & Smyth, 2001). We only included the causal relationships that had a .1+ probability value (Pr > .1) and can be prominent for the outcome variables. Predefined outcome variables for the analysis were two types of learning engagement (cognitive engagement and content engagement) together with subordinate modifiers for each type of learning engagement (Ke et al., 2015).

3. FINDINGS AND IMPLICATIONS

As shown by Figure 2, the sequential data mining results indicated that all major game actions promoted two types of learning engagement (i.e., cognitive and content engagement) differently. *Accurate Trading* action in the game was more likely to provoke knowledge *Application* (Pr =.329) and information *Processing* (Pr =.278) states of the content engagement. *Appropriate Allocation* action was another salient event to foster *Application* in content engagement (Pr = .4). On the other hand, *Positioning* move of the *Building* action showed a high probability (Pr = .467) in initiating the *Testing and Refining* state in cognitive engagement.

The study indicated the feasibility of using the combination of behavior analysis and sequential data mining for the research of game-based learning. The study findings suggested that learning game designers should purposefully design, evaluate, and select game events and actions that better promote learning engagement.

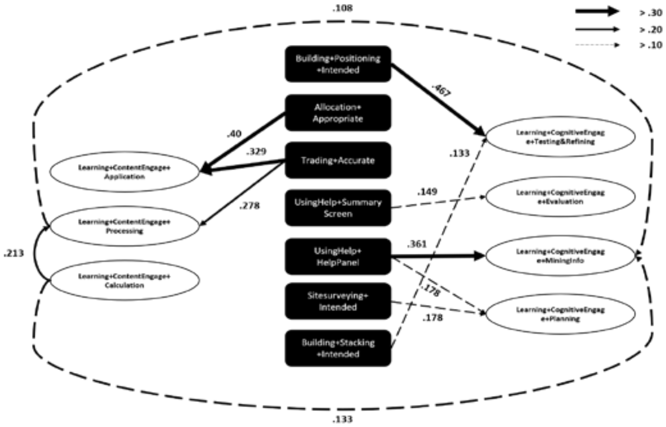

.108
> .30
> .20
> .10
Building+Positioning +Intended
Allocation+ Appropriate
Trading+Accurate
UsingHelp+Summary Screen
UsingHelp+ HelpPanel
Sitesurveying+ Intended
Building+Stacking +Intended
Learning+ContentEngage+ Application
Learning+ContentEngage+ Processing
Learning+ContentEngage+ Calculation
Learning+CognitiveEngag e+Testing&Refining
Learning+CognitiveEngag e+Evaluation
Learning+CognitiveEngag e+MiningInfo
Learning+CognitiveEngag e+Planning
.467
.40
.329
.278
.133
.149
.361
.178
.178
.213
.133

Figure 2. Emission probability values of game interactions to promote two types of learning engagement**.**

4. ACKNOWLEDGMENTS

This project is funded by the National Science Foundation, grant 1318784. Any opinions, findings, and conclusions or recommendations expressed in these materials are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

5. REFERENCES

[1] Friard, O., & Gamba, M. (2016). BORIS: a free, versatile open-source event-logging software for video/audio coding and live observations. *Methods in Ecology and Evolution, 7*(11), 1325-1330.

[2] Gottman, J. M., & Roy, A. K. (1990). *Sequential Analysis: A Guide for Behavorial Researchers*: Cambridge University Press.

[3] Hand, D. J., Mannila, H., & Smyth, P. (2001). *Principles of Data Mining*: CogNet.

[4] Hou, H.-T. (2015). Integrating cluster and sequential analysis to explore learners' flow and behavioral patterns in a simulation game with situated-learning context for science courses: A video-based process exploration. *Computers in Human Behavior, 48*(Supplement C), 424-435.

[5] Martin, T., & Sherin, B. (2013). Learning analytics and computational techniques for detecting and evaluating patterns in learning: An introduction to the special issue. *Journal of the Learning Sciences, 22*(4), 511-520.

[6] Ke, F. (2016). Designing Intrinsic Integration of Learning and Gaming Actions in a 3D Architecture Game. In Zheng, R., & Gardner, M. K. (Eds.), *Handbook of Research on Serious Games for Educational Applications*, (pp. 234-252). Hershey, PA: IGI Global.

Early Prediction of Course Grades: Models and Feature Selection

Hengxuan Li
Washington University in St. Louis
h.li@wustl.edu

Collin F. Lynch
North Carolina State University
cflynch@ncsu.edu

Tiffany Barnes
North Carolina State University
tmbarnes@ncsu.edu

ABSTRACT

In this paper, we compare predictive models for students' final performance in a blended course using a set of generic features collected from the first six weeks of class. These features were extracted from students' online homework submission logs as well as other online actions. We compare the effectiveness of 5 different ML algorithms (SVMs, Support Vector Regression, Decision Tree, Naive Bayes and K-Nearest Neighbor). We found that SVMs outperform other models and improve when compared to the baseline. This study demonstrates feasible implementations for predictive models that rely on common data from blended courses that can be used to monitor students' progress and to tailor instruction.

Keywords

Predictive model, machine learning, blended course, generic feature, support-vector machines

1. INTRODUCTION

In recent years, universities have begun to employ more online educational tools such as e-Textbooks, forums and homework submission systems. Online homework platforms, such as Webassign, support students and instructors by providing opportunities for automated grading and feedback. They can also support real-time monitoring of students' progress in the course.

If we can observe students' progress as they work and reliably predict their final grades, then can tailor the support provided to their needs. If, for example, a student's behavior indicates that they will succeed then simple encouragement (e.g. "keep up the good work") may be all that is required. If, however they are likely to fail, then they can be flagged for individual tutoring. Or they can be provided with automated guidance to useful resources or additional practice opportunities.

Our goal is to develop an accurate early predictor of students' final course grades from their user-system interaction logs. In order to strike a balance between early intervention and prediction accuracy. we trained our predictors based on the first 6 weeks of our 14-week course.

A number of researchers have sought to apply machine learning to predict students' course performance. Li et al. [9] proposed composite machine learning models based on features derived from students' interactions with forums, lectures, and assignments to identify at-risk students, and found that a Stacked Sparse Autoencoder+Softmax model achieved best AUC score consistently. Jiang et al. [8] sought to predict whether students would receive a completion certificate in a MOOC and if so what level it would be. To that end he combined their week 1 assignment performance with their online social interactions via logistic regression. They achieved 92.6% accuracy. Lopez et al. [10] applied a range of clustering methods to predict students' final marks in an online course based on their forum participation. They compared Expectation-Maximisation (EM) clustering, XMeans, Simple KMeans, and DTNB, using a set of four textual attributes and two network attributes: messages sent per student, replies per student, number of words written and the average expert rating of each message as well as the student's centrality and level of prestige within the social network. They found that the EM algorithm had higher accuracy than the alternatives. Similarly, Agnihotri et al. [1] applied K-Means clustering to login data from a web-based assessment platform called Connect and found a strong correlation between students' login patterns (e.g. opening assignments / attempting questions) and their scores. Brooks et al. [3] built a predictive model from time-series logs of student interactions with an online learning platform including quiz attempts, lecture views and posting to the forum. They used a decision tree to predict the students' final marks based on counting the different types of interactions over different time frames. Sabourin et al. [11] combined decision trees and Logistic Regression to classify students' self-regulated learning behaviors on an existing computer-based platform called Crystal Island. They found a weighted-by-Precision model to be most successful in classifying students' level of Self-Regulated Learning ("the process by which students activate and sustain cognitive, behaviors, and affects that are systematically directed toward the attainment of goals" [12]) performance (low, medium, high) through self-report prompts in game.

Bydzovska [5] evaluated multiple approaches to identify unsuccessful students. One approach used Support Vector machines (SVMs) and regression models based on social metrics, including measures of the students' betweenness and centrality (how many paths between students go through them). The other used collaborative filtering based on similarities between students' prior achievements. He found that the first approach reaches significantly better results for courses with a small number of students. In contrast, the second approach achieves significantly better results for mathematical courses. Stapel et al. [13] incorporated Knowledge Tracing with traditional machine learning such as K-Nearest Neighbor (KNN) and Naive Bayes to build an ensemble method to predict students' performance over specific math objectives, achieving an accuracy of 73.5%.

Holsta et al. [7] built a classification model to identify at-risk students without legacy data from other courses. This model used students' demographic, registration information in combination with online activity logs such as clicks in forum or assignment submissions. They compared the performance of these models on seven different datasets and found that XGBoost performanced better on average than SVMs, Linear Regression, KNN and Random Forests. Bote-Lorenzo et al. [2] found that Stochastic

Gradient Descent outperformed Linear Regression, SVMs and Random Forest at predicting the decrease of engagement of the students in a MOOC using a combination of assignment grades and submission statistics.

While most prior research was based on comparing the accuracy of different machine learning methods, the models used were either based on traditional onsite courses or MOOCs, and the number of features used was limited. In this paper, we built a model using generic features on homework submissions that are not unique to any specific course. We tested the accuracy of 5 different machine learning algorithms on data collected from a blended course, which pairs in-person lectures and office hours with an array of online tools including discussion forums, intelligent tutoring systems, and homework helpers. We employed Leave-One-Out cross validation to compare the accuracy of the different algorithms.

2. DATASET & FEATURES

We analyzed student data from CSC226 "Discrete Mathematics for Computer Scientists", a course offered by the Department of Computer Science at North Carolina State University. This is an introductory course for Computer Science (CS) and Computer Engineering students. It covers logic proofs, probability, set theory, combinatorics, graph theory, and finite automata. The dataset was collected from the Spring 2013 offering. This course has 2 lecture sections meeting 3 times per week, with 249 students total. The course lasted one semester (14 weeks) with 10 homework assignments, 2 intelligent tutors as labs and 4 tests (including the final). The final grade was based on the test scores (60%) and on the homework and lab assignments (40%). The final grade distribution is shown in Figure 1.

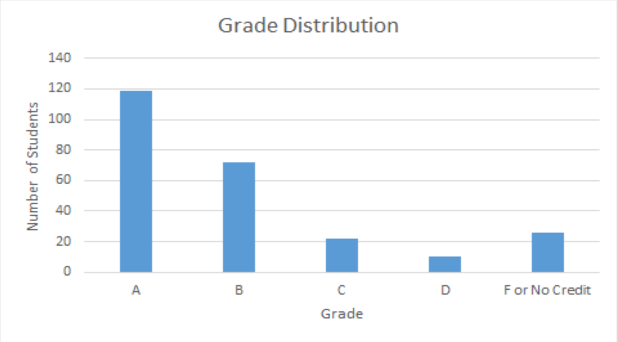
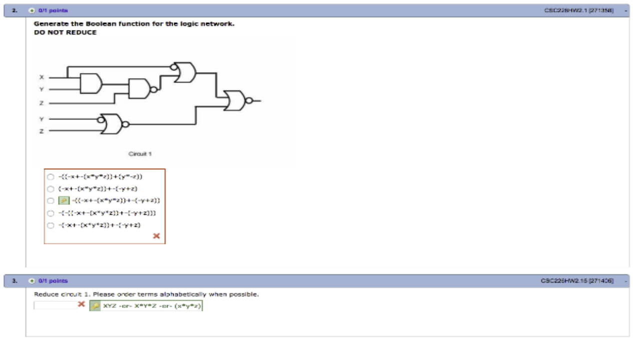


Figure 1: The grade distribution of course analyzed

We designed and compared a series of predictors based on the students' first 6 weeks of the coursework that includes four homework assignments and one test. The students completed their homework on Webassign, an online platform that supports automated grading and multiple retries. The homework questions were structured as short answer, fill in the blank (including Boolean values), or multiple choice questions. Complex questions such as the logic circuit shown in Figure 2, were broken into multiple submissions.

The students were typically given 1 attempt for each Boolean question and 3 attempts for all others. Our final dataset included 409 distinct questions with 265,510 submission attempts overall. The submission time was recorded as well as the student's section. The offline test was completed on paper as part of the students' class session and includes multiple open-ended questions. The test was graded manually. Homework and test scores are floating numbers between 0 and 100, inclusive.



2. 0/1 points CSC226HW2.1 [271358]

Generate the Boolean function for the logic network.
DO NOT REDUCE

○ ~((~x+~(x*y*z))+(y*~z))
○ (~x+~(x*y*z))+~(~y+z)
○ ~((~x+~(x*y*z))+~(~y+z))
○ ~(~((~x+~(x*y*z))+~(~y+z)))
○ ~(~x+~(x*y*z))+~(~y+z)

3. 0/1 points CSC226HW2.15 [271406]

Reduce circuit 1. Please order terms alphabetically when possible.

XYZ -or- X*Y*Z -or- (x*y*z)

Figure 2: A sample question on Webassign

3. METHODS

We represented student performance with the features to represent shown in Table 1.

3.1 Feature Selection

We used the VarianceThreshold method from the Sci-Kit Learn Python library (version 0.19.0) to perform feature selection. The dataset includes some easy questions that almost every student answered correctly in one submission, indicating the corresponding features did not have much variance. Therefore, no good predictions can be made from these features, so we eliminated them from analysis. To save computing power and avoid spurious correlation between these features, we tested several combinations of thresholds of variance. The combinations tested for Per-Question Performance and Submissions Per Question respectively were (0.00, 0.00), (0.02,0.05), (0.03, 0.07), (0.04, 0.10). By checking the final accuracy of predictors after running the models under different thresholds, we found (0.02, 0.05) achieved the best accuracy. Therefore, we chose (0.02, 0.05) as the threshold in our analysis, and it selects 311 and 329 features for Per-Question Performance and Submissions Per Question, where original number of features are all 409.

Table 1: The feature list

Feature	Total
Per-Question Performance: Whether a student answers questions correctly indicating skill mastery.	409
Submissions Per Question: Number of tries per question indicating the number of errors or guesses.	409
Response Time: Extra-long response times indicate that the student may be distracted or having difficulty with the question while very short response times may indicate guesses. Long responses are defined as response times two standard deviations above average while quick response times are > 5 per minute. We exclude responses that are longer than 2 hours as this indicates a disconnected session.	4
Sessions Per Assignment: A session is defined as a period of time taken on homework. Two adjacent tries within 2 hours are treated as the same session. Multiple sessions per assignment may indicate difficulty with the assignment.	4
Homework and Test Scores: are used in calculating the final grade.	5

3.2 Normalization and Manual Segmentation

After eliminating uninformative features, we normalized each value to the range [0,1] to prevent any one feature from dominating the others. We then compared the performance of our trained models on both the normalized and unnormalized data to assess the impact of this step.

We also plotted the distribution of the submission attempts and response times for each question in order to assess their utility. Both distributions are dramatically right-skewed. Therefore, we did not expect manual segmentation from the decision tree to be more meaningful than the automatic segmentation provided by the Sci-Kit library. Therefore, we therefore opted not to perform any manual segmentation in this study.

3.3 Machine Learning and Cross Validation

We used the following standard implementations of the machine learning methods from the Sci-Kit library to train our models: Support Vector Machine (RBF kernel), Support Vector Regression, Decision Tree (Scikit-Learn uses an optimized version of the CART algorithm.), Naive Bayes and K-Nearest Neighbors (K=5). In order to assess the performance of the trained models, we also added two baseline models: random prediction and predicting the most frequent grade (A in this case).

We then estimated the stability of the models using Leave-One-Out cross validation we report the overall accuracy and a confusion matrix for each algorithm along with an, average precision score (micro over cross-validation), AUROC (exactly correct vs. not exactly correct), f1 score and mean squared error is also calculated to better compare the performances of models.

4. RESULTS

Because Support Vector Machine and Support Vector Regression use regularization (C=1.0) to prevent overfit, and the other models are sensitive to changes in attributes' values, the normalization process impacted performance. However, it did not lead to any consistent improvement in the accuracy relative to the non-normalized models. Because the accuracy of the Support Vector Machine and Linear Regression methods dropped significantly after normalization, we will focus solely on the non-normalized models in the remainder of the paper.

Based on Leave-One-Out cross validation, Support Vector Machines perform best among all the five algorithms, achieving 54.1% accuracy. Support Vector Regression, Decision Tree and K-Nearest neighbor reached more than 40% accuracy, but Naive Bayes performed just slightly above chance. The other performance statistics showed the same trend.

Table 2: Performance for non-normalized input

	Accuracy	Mean Square Error	Average Precision (Micro)	AUROC	f1 score
SVM	51.4%	1.755	0.36	0.573	0.514
Lin. Reg	45.8%	1.304	0.23	0.558	0.289
Decision Tree	43.8%	1.803	0.3	0.590	0.437
Naive Bayes	24.1%	3.108	0.21	0.539	0.240
KNN	41.8%	1.510	0.29	0.595	0.417
Random	20.0%	4.807	0.2	0.492	0.196
All A	47.8%	2.674	0.33	0.500	0.477

Table 3: Performance for normalized input

	Accuracy	Mean Square Error	Average Precision (Micro)	AUROC	f1
SVM	51.0%	2.160	0.36	0.536	0.510
Lin. Reg	23.7%	1.459	0.23	0.523	0.301
Decision Tree	42.2%	1.702	0.29	0.576	0.421
Naive Bayes	25.3%	3.108	0.21	0.543	0.253
KNN	24.1%	1.767	0.21	0.554	0.240
Random	20.0%	4.807	0.2	0.492	0.196
All A	47.8%	2.674	0.33	0.500	0.477

We then generated confusion matrices for the different approaches. These matrices are shown in Tables 4 & 5. Here the difference is the absolute distance between the predicted grade and the actual grade on an integer scale (5-A 4-B 3-C 2-D 1-F)

Table 4: Confusion matrix for unnormalized input

	0	1	2	3	4
SVM	128	81	19	8	13
Lin. Reg	114	89	34	7	5
Decision Tree	109	94	23	15	8
Naive Bayes	60	82	78	12	17
KNN	104	98	36	6	5
All predict to A	119	72	22	10	26

Table 5: Confusion matrix for normalized input

Difference	0	1	2	3	4
SVM	127	72	22	10	18
Lin. Reg	59	159	13	16	2
Decision Tree	105	98	26	14	6
Naive Bayes	63	78	79	12	17
KNN	60	116	68	4	1

5. DISCUSSION & FUTURE WORK

While the machine learning models described in this study can be used to predict students' final grades to some extent, the accuracy is still far from ideal for real-world applications. Although the best model (SVMs) performed better than the naive baseline models, the advantage is not significant. At the same time, normalization did not bring us any notable improvement. When examining the misclassified students, we found that a considerable portion of students who did well in the homework actually performed poorly in the first test. Given the high percentage (47.8%) of A grades in this course and the fact that homework typically permitted multiple tries we concluded that the homework may have been too easy, and that students' final homework scores were not reliable predictors of their future test scores, which are in turn the largest portion of final score. Thus, it was not possible to derive a good predictive model that relies heavily on homework submission logs. We also noticed that almost all of the students who did not complete or performed poorly in one of the assignments eventually dropped the course. We believe that these are students who may have wanted to drop the course and who thus quit doing the homework before dropping or who were motivated to do so after a particularly bad homework score. Unfortunately, none of the models correctly captured this phenomenon.

In the future, we hope to examine if feature engineering can be used to address the limitations above. If we can predict dropouts in advance, then we can make the models much more robust. One other possible way to improve upon this is to add additional features. A richer model may be more robust in the face of noise. Combining this interaction model with models based on social network data, for example, may improve our performance particularly in cases where help-seeking is an important indicator of performance. Brown et. al [4] have shown that students on MOOCs formed detectable communities, and community membership was significantly correlated with performance. In addition, Gitinabard et. al [6] showed that students who asked more questions and received more feedback on the forum tended to obtain higher grades in blended courses. It will be interesting to see if students closely connected in a social network in course influence each other and further change the homework pattern of features overtime.

ACKNOWLEDGMENTS

The authors wish to thank Zhongxiu Aurora Liu and members of that Center of Educational Informatics at North Carolina State University for their assistance.

This research was partially supported by the National Science Foundation Grant #1418269: “Modeling Social Interaction & Performance in STEM Learning” Yoav Bergner, Ryan Baker, Danielle S. McNamera, & Tiffany Barnes Co-PIs.

6. REFERENCES

[1] Agnihotri, L., Aghababyan, A., Mojarad, S., Riedesel, M., and Essa, A. 2015. Mining Login Data For Actionable Student Insight. *Proceedings of the 8th International Conference on Educational Data Mining*. 472-475.

[2] Bote-Lorenzo, M., Gómez-Sánchez, E. 2017. Predicting the decrease of engagement indicators in a MOOC. *Proceedings of the 7th International Learning Analytics and Knowledge Conference.* 143-147.

[3] Brooks, C., Thompson, C., and Teasley S. 2015. A Time Series Interaction Analysis Method for Building Predictive Models of Learners using Log Data. *Proceedings of the 5th International Conference on Learning Analytics and Knowledge, ACM.* 126-135.

[4] Brown, R., Lynch, C., Eagle, M., Albert, J., Barnes, T., Baker, R., Bergner, Y. and McNamara, D. 2015. Good Communities and Bad Communities: Does membership affect performance? *Proceedings of the 8th International Conference on Educational Data Mining.* 612-613.

[5] Bydzovska H. 2016. A Comparative Analysis of Techniques for Predicting Student Performance. *Proceedings of the 9th International Conference on Educational Data Mining.* 306-311.

[6] Gitinabard, N., Xue L., Lynch, C., Heckman, S. and Barnes, T. 2017. A Social Network Analysis on Blended Courses. arXiv: 1709.10215. Retrieved from https://arxiv.org/pdf/1709.10215.pdf

[7] Holsta, M., Zdrahal, Z., Zendulka, J. 2017. Ouroboros: Early identification of at-risk students without models based on legacy data. *Proceedings of the 7th International Learning Analytics and Knowledge Conference.* 6-15.

[8] Jiang, S., Adrienne, Williams, A.E., Schenke, K., Warschauer, M., and O’Dowd, D. 2014. Predicting MOOC Performance with Week 1 Behavior. *Proceedings of the 7th International Conference on Educational Data Mining.* 273-275.

[9] Li, Y., Fu C., Zhang Y. 2017. When and who at risk? Call back at these critical points. *Proceedings of the 10th International Conference on Educational Data Mining.* 168-173.

[10] Lopez, M.I., Luna, J.M., Romero, C., and Ventura, S. 2012. Classification via clustering for predicting final marks based on student participation in forums. *Proceedings of the 5th International Conference on Educational Data Mining.* 148-151.

[11] Sabourin, J.L., Mott, B.W., and Lester, J.C. 2012. Early Prediction of Student Self-Regulation Strategies by Combining Multiple Models. *Proceedings of the 5th International Conference on Educational Data Mining.* 156-159.

[12] Schunk, D. H. Attributions as Motivators of Self-Regulated Learning, *in Motivation and Self-Regulated Learning: Theory, Research, and Applications, D. H. Schunk and B. J. Zimmerman (eds).* 2008, pp. 245-266.

[13] Stapel, M., Zheng, Z., and Pinkwart, N. 2016. An Ensemble Method to Predict Student Performance in an Online Math Learning Environment. *Proceedings of the 9th International Conference on Educational Data Mining.* 231-238.

Who creates artifacts in a MOOC on Yoga?

Sai Santosh Sasank Peri
LINK Research Lab
University of Texas at Arlington
1-817-272-3210
saisantoshsasan.peri@mavs.uta.edu

James D. Schaeffer
LINK Research Lab
University of Texas at Arlington
1-817-272-3210
James.Schaeffer@mavs.uta.edu

Catherine A. Spann
Institute of Cognitive Science
University of Colorado at Boulder
1-303-735-5790
catherine.spann@colorado.edu

Angela Liegey Dougall
Department of Psychology
University of Texas at Arlington
1-817-272-2281
adougall@uta.edu

George Siemens
LINK Research Lab
University of Texas at Arlington
1-817-272-3210
gsiemens@uta.edu

ABSTRACT

Learner-created artifacts are social learning objects that enable creativity, augment learning and reflect learners' thoughts. The purpose of this study was to identify psychosocial characteristics of learners that predict artifact creation. Learners (N = 1708) enrolled in a Massive Open Online Course (MOOC) called The Science and Practice of Yoga and created artifacts as part of their weekly activity. Learners used popular social media tools like blog posts, memes, inspirational posters, concept maps and animated gifs to create artifacts. The course design required learners to share their creative work with their peers. In the pre-course survey, learners self-reported personality traits, mindfulness, emotion regulation, psychological well-being and health. We divided the learners into two groups based on whether or not they posted course related artifacts and performed an independent samples *t*-test for each of the psychological scales. We observed that learners reporting higher scores on psychological and general health were more likely to create artifacts. We infer that these learners would have enrolled in this MOOC to use yoga as way to improve their physical and mental health while using artifacts as a channel to share their thoughts and connect with other students in the MOOC.

Keywords

Student-produced artifacts, psychosocial characteristics, MOOC, Knowledge space

1. INTRODUCTION

The increased use of social technologies in society is being reflected in formal and informal learning [1, 2]. Limited analysis has been conducted [3] regarding how learners in MOOCs use social media. In our study, we investigated the use of social media and artifact creation. Learners' usage of social media platforms and the survey responses in The Science and Practice of Yoga indicated that learning could be augmented by providing learners with an opportunity to share resources and communicate thoughts and reflections effectively.

Often, the conversations in social media include usage of tools and objects such as memes, animated gifs, URLs to blog posts or other web resources [3-5]. These tools give learners greater autonomy to express themselves and their conception of course content. These contributions may be related to the topic being discussed or completely detour and create a new conversation surrounding. We refer to these social objects as artifacts.

When artifacts are incorporated into online courses, they have potential to enhance learning [6]. These artifacts are learner-created and are social in nature – sharing how the learner interpreted course material and the connections made to existing knowledge. Previous research has examined different learning pathways and knowledge spaces involved in the usage of social media that could enhance the learning experience of students, rather than a traditional instructor driven pathway [7, 8]. Similarly, the creation of artifacts can serve as an alternative learning pathway and space for building knowledge.

In our MOOC, The Science and Practice of Yoga, learners were required to use social media tools, like blog posts, memes, inspirational posters, concept maps and animated gifs to create content that expressed their thoughts and reflections. For example, a learner could create an image that related to a certain concept of yoga and post it to the general discussion forum of the course. Other learners could share their reactions, critique the image, and even add to it. Such an activity creates an additional knowledge contribution that augments what was designed in the course.

Although a course might explicitly ask learners to create artifacts, not everyone will participate. Understanding who creates artifacts in a MOOC is an open, empirical question. To address this question, we grouped learners from the MOOC on The Science and Practice of Yoga based on whether or not they created artifacts and investigated psychological differences between these groups. We administered a number of validated psychological scales related to psychological health and well-being in order to investigate whether or not psychological factors could differentiate groups and predict artifact creation in our MOOC.

2. METHOD

2.1 Participants

Initially, 20347 learners signed up to take a 6-week MOOC on the science and practice of yoga. The course lasted for 6 weeks between October and December 2017. Of those learners who initially signed up, 3755 learners consented for the pre-course survey. 1708 learners completed the survey. Among those that completed the survey, 178 posted at least one course related artifact in the discussion forum of the MOOC.

2.2 Survey Measures

The pre-course survey was created to assess a variety of demographic and psychological variables. These variables were selected because we thought that they might either broadly relate to learner participation in course activities or specifically relate to activities in a MOOC about yoga and meditation. Demographic variables included age, gender, education, and a measure of income.

We assessed participants' self-efficacy (i.e. confidence in their own abilities), beliefs about the effectiveness of yoga and meditation, their intentions to perform yoga and meditation, prior yoga and meditation experience, and prior online course experience. We also assessed a number of mental and physical health variables using validated scales. The Patient-Reported Outcomes Measurement Information System (PROMIS) scale was used to measure physical and mental health, where higher scores represent better physical and health [9]. The Curiosity and Exploration Inventory-II (CEI) was used to measure *stretching* (motivation to seek new experiences) and *embracing* (willingness to embrace new situations); higher scores represent more of each [10]. The Emotion Regulation Questionnaire (ERQ) was used to measure the two emotion regulation strategies of *reappraisal* and *suppression* (reappraisal is believed to be a healthy strategy, whereas suppression is not); higher scores on each represent a greater tendency toward that regulation strategy [11]. The Five Facet Mindfulness Questionnaire (FFMQ) was used to measure the five elements of mindfulness including *observing*, *describing*, *acting with awareness*, *nonjudging of inner experiences*, and *nonreactivity of inner experiences*; higher scores for each represent greater mindfulness levels for each [12]. The 18-question version of the Psychological Well-being (PWB) Scale was used to measure six well-being elements including *self-acceptance*, *environmental mastery*, *positive relations with others*, *purpose in life*, *personal growth*, and *autonomy*; higher scores for each represent greater psychological well-being [13]. The 4-item Perceived Stress Scale (PSS) was used to measure perceived stress over the past month prior to course participation; higher scores represent greater stress [14]. The Sense-Of-Self Scale (SOSS) was used to measure understanding of one's self; higher scores represent a poorer understanding of one's self [15]. The Ten-Item Personality Inventory (TIPI) was used to measure the Big-5 personality constructs of *openness*, *conscientiousness*, *extraversion*, *agreeableness*, and *emotional stability*; higher scores represent a greater amount of each personality trait [16].

2.3 Procedure

At the beginning of the MOOC, learners were asked to complete an optional pre-course survey involving the above-mentioned scales, which was administered by Qualtrics through EdX. The survey took approximately 45 minutes to complete. Learner mental and physical health data were collected throughout the 6 weeks.

While learners were participating in the MOOC they had access to the course discussion forum where they could interact with other learners and instructors. The learners were required to post the course related artifacts in this space. The artifacts (memes, blog posts, gif, etc.) had a URL associated with it, which was recorded along with all of the discussion forum posts.

Learner's discussion forum posts were collected from the EdX data repository of the course. The posts containing artifacts were filtered out. A unique list of learners who posted at least one course related artifact was created. Only learners who completed the pre-survey were considered. A final data set was created containing the learners who completed the pre-survey, their survey results and a binary value was given to each learner based on whether or not they posted a course related artifact. This dataset was used for investigation of psychological and health differences between the two groups.

2.4 Analysis of Survey Data

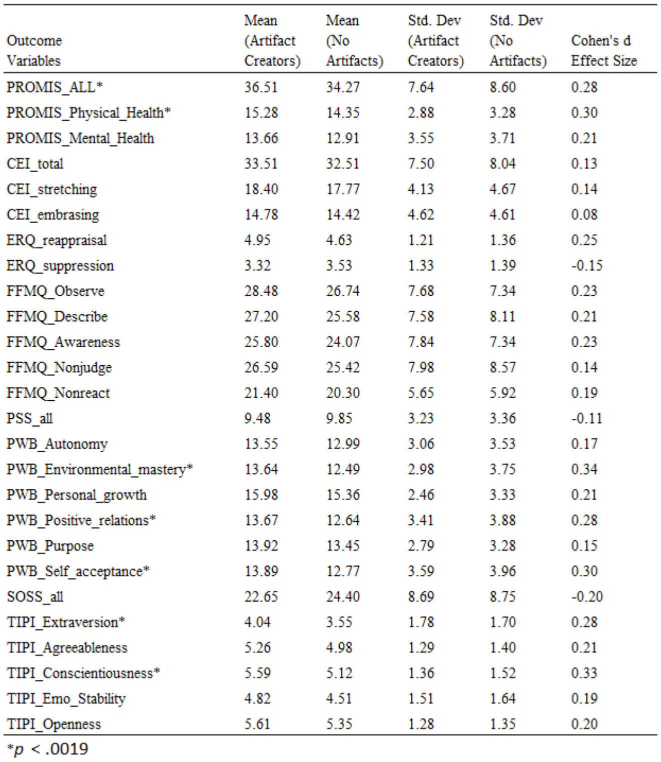
Using SPSS statistical software, we used Independent Samples T-tests to examine differences in mental and physical health between those who created artifacts and those who did not. Outcome variables included all the psychological scales from the pre-survey.

3. RESULTS

The mean, standard deviation and effect size values of all outcome variables are reported in table 1. Our alpha value over 26 tests was calculated to be .0019.

On the PROMIS scale, artifact creators reported to be significantly phsycially healthier than learners' who had no artifacts. t(1706) = 3.593, p < .001 and represented a moderate-size effect, d = 0.3. On the PWB scales, we oberserved that artifact creators had significantly greater scores on environmental mastery (t(1706) = 3.961, p < .001), postive relations (t(1706) = 3.394, p = .001) and self acceptance (t(1706) = 3.603, p < .001) than rest of the learners. These three PWB scales represented a small to moderate size effect with Cohen's d value of 0.34, 0.28, 0.30 respectively.

Table 1: Means, SDs, and effect sizes (Cohen's d) for outcome variables across artifact creators (*n*=178) and no artifacts (*n*=1530)



Outcome Variables	Mean (Artifact Creators)	Mean (No Artifacts)	Std. Dev (Artifact Creators)	Std. Dev (No Artifacts)	Cohen's d Effect Size
PROMIS_ALL*	36.51	34.27	7.64	8.60	0.28
PROMIS_Physical_Health*	15.28	14.35	2.88	3.28	0.30
PROMIS_Mental_Health	13.66	12.91	3.55	3.71	0.21
CEI_total	33.51	32.51	7.50	8.04	0.13
CEI_stretching	18.40	17.77	4.13	4.67	0.14
CEI_embrasing	14.78	14.42	4.62	4.61	0.08
ERQ_reappraisal	4.95	4.63	1.21	1.36	0.25
ERQ_suppression	3.32	3.53	1.33	1.39	-0.15
FFMQ_Observe	28.48	26.74	7.68	7.34	0.23
FFMQ_Describe	27.20	25.58	7.58	8.11	0.21
FFMQ_Awareness	25.80	24.07	7.84	7.34	0.23
FFMQ_Nonjudge	26.59	25.42	7.98	8.57	0.14
FFMQ_Nonreact	21.40	20.30	5.65	5.92	0.19
PSS_all	9.48	9.85	3.23	3.36	-0.11
PWB_Autonomy	13.55	12.99	3.06	3.53	0.17
PWB_Environmental_mastery*	13.64	12.49	2.98	3.75	0.34
PWB_Personal_growth	15.98	15.36	2.46	3.33	0.21
PWB_Positive_relations*	13.67	12.64	3.41	3.88	0.28
PWB_Purpose	13.92	13.45	2.79	3.28	0.15
PWB_Self_acceptance*	13.89	12.77	3.59	3.96	0.30
SOSS_all	22.65	24.40	8.69	8.75	-0.20
TIPI_Extraversion*	4.04	3.55	1.78	1.70	0.28
TIPI_Agreeableness	5.26	4.98	1.29	1.40	0.21
TIPI_Conscientiousness*	5.59	5.12	1.36	1.52	0.33
TIPI_Emo_Stability	4.82	4.51	1.51	1.64	0.19
TIPI_Openness	5.61	5.35	1.28	1.35	0.20

*p < .0019

On the TIPI scale, artifact creators self-reported to have significantly higher scores on personality traits like extraversion (t(1706) = 3.59, p < .001) and conscientiousness (t(1706) = 3.96, p < .001) representing a small to medium size effect with Cohen's d value of 0.28, 0.33 respectively.

4. DISCUSSION

A higher score on physical health indicates artifact creators were higher functioning and had a better quality of life, before taking the course. Artifact creators exhibited higher environmental mastery, which meant they are always in charge of any situation in their lives. They also had higher score on positive relations with others, which shows that they were willing to share time with their peers through creation of artifacts and indulging in conversations surrounding it. They had a good self-acceptance score meaning that they had a positive attitude towards themselves.

Artifact creators were outgoing and energetic (extraversion) while being efficient and organized (conscientiousness) all together. We can infer that these set of learners were self-disciplined, work towards a goal and have a tendency to lead in a social environment. All these characteristics describe their active participation in all the artifact creation activities of the MOOC. They were likely to use artifacts as a channel to share their thoughts and connect with other students in the MOOC.

While additional investigation is required, these results suggest that an active and engaged approach to learning – creating rather than consuming – may be related to psychological attributes that are currently not well understood in MOOC literature. For example, active learning in classrooms contributes to significantly better learning outcomes than lecture-based learning [17]. There is reason to believe that artifact creation, as a form of active learning, produces similar learning gains. Additionally, questions exist regarding the ability to shape psychological attributes, through course design or teaching practices, to involve currently passive learners in artifact creation.

5. Conclusion

We showed that learners who posted course related artifacts could be differentiated from those who do not based on underlying psychological characteristics. Primarily, we showed that students who posted artifacts were generally mentally and physically healthier than those who did not post artifacts. Our findings suggest that underlying psychological factors may influence student performance in MOOCs. However, it is important to note that the effect sizes in the present study were small to moderate in size. Although psychological factors may be able to discriminate these groups, other unobserved factors could be playing a much larger role. Further research is required to investigate group differences and increase our understanding of student artifact creation and active learning in MOOCs.

6. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Award Number 1546393.

7. REFERENCES

[1] John B. Horrigan. 2016. The joy – and urgency – of learning. (March 2016). Retrieved April 24, 2018 from http://www.pewinternet.org/2016/03/22/the-joy-and-urgency-of-learning/

[2] Greenhow, C. and Lewin, C. Social media and education: reconceptualizing the boundaries of formal and informal learning. Learning, media and technology, 41, 1 (2016), 6-30.

[3] Liu, M., McKelroy, E., Kang, J., Harron, J. and Liu, S. Examining the use of Facebook and Twitter as an additional social space in a MOOC. *American Journal of Distance Education*, 30, 1 (2016), 14-26.

[4] Mae Duggan. 2013. Photo and Video Sharing Grow Online. (October 2013). Retrieved April 24, 2018 from http://www.pewinternet.org/2013/10/28/photo-and-video-sharing-grow-online/

[5] Amanda Lenhart, Mary Madden, Aaron Smith and Alexandra Macgill. 2007. Teens creating content. (December 2007). Retrieved April 24, 2018 from http://www.pewinternet.org/2007/12/19/teens-creating-content/

[6] Downes, S. Places to go: Connectivism & connective knowledge. Innovate: Journal of Online Education, 5, 1 (2008), 6.

[7] Friedman, L. W. and Friedman, H. H. Using social media technologies to enhance online learning. Journal of Educators Online, 10, 1 (2013), n1.

[8] Crosslin, M. and Dellinger, J. Lessons learned while designing and implementing a multiple pathways xMOOC+ cMOOC. Association for the Advancement of Computing in Education (AACE), City, 2015.

[9] Hays, R.D., Bjorner, J.B., .Revicki, D.A., Spritzer, K.L., & Cella, D. (2009). *Development of physical and mental health summary scores from the patient-reported outcomes measurement information system (PROMIS) global items.* Quality of Life Research, 18, 873-880.

[10] Kashdan, T.B., Gallagher, M.W., Silvia, P.J., Winterstein, B.P., Breen, W.E., Terhar, D., & Steger, M.F. (2009). *The curiosity and exploration inventory-II: Development, factor structure, and psychometrics*. Journal of Research in Personality, 43, 987-998.

[11] Gross, J.J. & John, O.P. (2003). *Individual differences in two emotion regulation processes: Implications for affect, relationships, and well-being.* Journal of Personality and Social Psychology, 85, 348-362.

[12] Baer, R. A., Smith, G. T., Hopkins, J., Krietemeyer, J., & Toney, L. (2006). *Using self-report assessment methods to explore facets of mindfulness.* Assessment, 13, 27-45.

[13] Ryff, C.D., & Keyes, C.L.M. (1995). *The structure of psychological well-being revisited*. Journal of Personality and Social Psychology, 69(4), 719-727.

[14] Cohen, S., Kamarck, T., & Mermelstein, R. (1983). A global measure of perceived stress. Journal of health and Social Behavior, 24, 385-396.

[15] Flury, J.M. & Ickes, W. (2007). *Having a weak versus strong sense of self: The sense of self scale (SOSS).* Self and Identity, 9, 281-303.

[16] Gosling, S.D., Rentfrow, P.J., & Swann Jr., W.B. (2003). *A very brief measure of the big-five personality domains.* Journal of Research in Personality, 37, 504-528.

[17] Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H. and Wenderoth, M. P. Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111, 23 (2014), 8410-8415.

Learner subpopulations in massive open online courses differ by psychological and demographic variables: A discriminant function analysis

James D. Schaeffer
LINK Research Lab
University of Texas at Arlington
1-817-272-3210
James.Schaeffer@mavs.uta.edu

Sai Santosh Sasank Peri
LINK Research Lab
University of Texas at Arlington
1-817-272-3210
saisantoshsasan.peri@mavs.uta.edu

Catherine A. Spann
Institute of Cognitive Science
University of Colorado at Boulder
1-303-735-5790
catherine.spann@colorado.edu

Angela Liegey Dougall
Department of Psychology
University of Texas at Arlington
1-817-272-2281
adougall@uta.edu

George Siemens
LINK Research Lab
University of Texas at Arlington
1-817-272-3210
gsiemens@uta.edu

ABSTRACT

There has been much interest in developing profiles of learners engaging in Massive Open Online Courses (MOOCs) in order to better predict learner behavior and maximize learning outcomes. Recent work by Kizilcec, Piech, and Schneider (2013) showed that weekly patterns of learner engagement with course materials could be used to cluster learners into four groups: completing, auditing, disengaged, and sampling. In the present study, we sought to understand the characteristics of these learners by investigating demographic and psychological variables collected from a pre-course survey in an EdX MOOC called The Science and Practice of Yoga. First, we employed hierarchical and K-means clustering on weekly engagement patterns in the MOOC and clustered learners into highly similar groups as in Kizilcec et al., thereby replicating their findings. Next, we employed principal component analysis and discriminant function analysis to compare demographic and psychological variables. Principal component analysis suggested three categories: mental health, self and course beliefs, and curiosity and openness. Discriminant function analysis was able to discriminate groups based on these variables. Function 1 separated Completing Learners from the rest, and Function 2 separated Disengaged Learners from the rest. These findings suggested that engagement patterns in MOOCs might be partly explained by learners' psychological traits and pre-course states. This has implications for how MOOCs are designed to foster planned interactions that learners have with one another and with the course content by advancing consideration of learner psychological attributes, rather than primarily the content to be learned.

Keywords

MOOC, Learner Profiles, Engagement

1. INTRODUCTION

Massive open online courses (MOOCs) continues to gain global attraction. The large data sets generated have proven useful for learning analytics, educational data mining, and learning sciences in general. The analysis of these data sets has to date largely focused on content and learner interactions, with minimal attention paid to psychological attributes of learners. In our study, we begin to address this gap. In 2013, Kizilzec and others developed a clustering method for organizing learners into groups based on their participation and engagement patterns in MOOCs. In their study [1], engagement was determined by their participation and assignment completion rates. Engagement was measured on a weekly basis and summed to create a score used for clustering. For each week, learners were characterized as "on track" (completed work on time), "behind" (completed work late), "auditing" (didn't complete work, but engaged by watching videos), or "out" (didn't participate at all), and were given a score of 3 for on track, 2 for behind, 1 for auditing, and 0 for out. At the end of the course, these scores were summed and the summed values were used for clustering.

In the present study, we replicated this method using learner participation and engagement patterns in the EdX MOOC, The Science and Practice of Yoga. We also sought to investigate psychological differences between these groups using a pre-course survey. We administered a number of validated psychological scales related to psychological health and well-being in order to investigate whether or not psychological factors could differentiate groups and predict participation and engagement patterns in our MOOC.

2. METHOD

2.1 Participants

Initially, 20347 learners signed up to take a 6-week MOOC on the science and practice of yoga. Of those learners who initially signed up, 3755 learners consented to participate in the study and completed a pre-course survey. These 3755 were included in the analyses.

2.2 Survey Measures

The pre-course survey was created to assess a variety of demographic and psychological variables. These variables were selected to assess if they either broadly related to learner engagement or specifically related to engagement in a MOOC about yoga and meditation. Demographic variables included age, gender, education, and a measure of income.

Psychological variables were measured through scores on scales created by the authors, including those designed to assess participants' self-efficacy (i.e. confidence in their own abilities), beliefs about the effectiveness of yoga and meditation, their intentions to perform yoga and meditation, prior yoga and meditation experience, and prior online course experience. We also assessed a number of psychophysiological health variables using validated scales. The Patient-Reported Outcomes Measurement Information System (PROMIS) scale was used to measure physical and mental health; higher scores represent better physical and health [2]. The Curiosity and Exploration Inventory-II (CEI) was used to measure *stretching* (motivation to seek new experiences) and *embracing* (willingness to embrace new situations); higher scores represent more of each [3]. The Emotion Regulation Questionnaire (ERQ) was used to measure the two emotion regulation strategies of *reappraisal* and *suppression* (reappraisal is believed to be a healthy strategy, whereas suppression is not); higher scores on each represent a greater tendency toward that regulation strategy [4]. The Five Facet Mindfulness Questionnaire (FFMQ) was used to measure the five elements of mindfulness including *observing*, *describing*, *acting with awareness*, *nonjudging of inner experiences*, and *nonreactivity of inner experiences*; higher scores for each represent greater mindfulness levels for each [5]. The 18-question version of the Psychological Well-being (PWB) Scale was used to measure six well-being elements including *self-acceptance*, *environmental mastery*, *positive relations with others*, *purpose in life*, *personal growth*, and *autonomy*; higher scores for each represent greater psychological well-being [6]. The 4-item Perceived Stress Scale (PSS) was used to measure perceived stress over the past month prior to course participation; higher scores represent greater stress [7]. The Sense-Of-Self Scale (SOSS) was used to measure a lack of understanding of one's self; higher scores represent a poorer understanding of one's self [8]. The Ten-Item Personality Inventory (TIPI) was used to measure the Big-5 personality constructs of *openness*, *conscientiousness*, *extraversion*, *agreeableness*, and *emotional stability*; higher scores represent a greater amount of each personality trait [9].

2.3 Procedure

Survey data and engagement patterns were collected from participants in the Science and Practice of Yoga MOOC. The course lasted for 6 weeks between October and December 2017. Learners were asked to complete an optional pre-course survey involving the abovementioned scales, which was administered by Qualtrics through EdX. The survey took roughly 30 minutes to complete (95% trimmed mean of 30.7 minutes). Engagement data were collected throughout the 6 weeks, and included video viewing and quiz completion throughout those 6 weeks.

Clustering based on learner engagement was conducted following the procedures outlined in Kizilcec et al. [1] based on learner engagement patterns. Like Kizilcec et al., learners were given a score between 0 and 3 for each week. Learners considered *on track* received a 3, learners considered *behind* received a 2, learners considered *auditing* received a 1, and learners considered *out* received a 0. These scores were summed and then used for clustering. However, our MOOC requirements were slightly different from those in Kizilcec et al. We did not have weekly assignments, but instead had weekly quizzes that did not have a deadline. Therefore, our weekly engagement patterns were measured slightly differently. Learners were classified as *on track*, *behind*, *auditing*, or *out* each week based on when they completed their weekly quizzes. Quiz completion times were z-transformed. Learners were considered *on track* if they completed their assignments earlier than the mean time, and learners were considered *behind* if they completed their assignments later than the mean time. *Auditing* learners did not complete the quiz at all, but still watched the weekly videos. *Out* learners did not watch any video or complete any quiz for that week. Data were clustered initially using hierarchical clustering, where the order of the data were randomized. Clustering of data were conducted on the 3755 learners who completed the pre-course survey. For each, dendrograms from hierarchical centroid clustering based on squared Euclidean distances suggested that 4 clusters could be a viable option for our data (Figure 1). K-means clustering with 4 solutions was used to create the 4 groups of learners reported in Kizilcec et al. (i.e. Completing, Auditing, Disengaged, and Sampling learners).

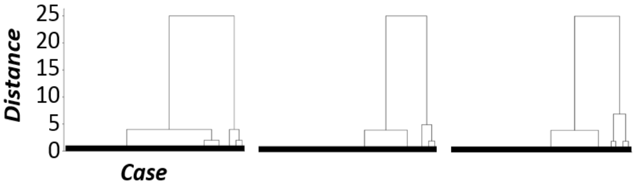


Figure 1. Three dendrograms from hierarchical cluster analyses. Data were randomly sorted for each

2.4 Analysis of Survey Data

Initially, principal component analysis was used on the individual scales of the survey to determine unique components in our dataset. Because many of the survey measures included relate to similar psychological constructs, we used PCA to reduce the number of measures prior discriminant function analysis. The variables making up each of these three components were then entered into a discriminant function analysis designed to discriminate based on group clusters (i.e. completing, auditing, disengaged, or sampling learners).

3. RESULTS

3.1 Clustering

K-means clustering results largely matched the results of Kizilcec et al. Learners were clustered into 4 groups: Completing learners (those who completed most of the quizzes and videos throughout the course), Auditing learners (those who were engaged consistently throughout the course but did not often complete the quizzes), Disengaged learners (those who initially showed high engagement, and then dropped off sharply later in the course), and Sampling learners (those who watched a video every now and then). These findings are presented in Figure 2.

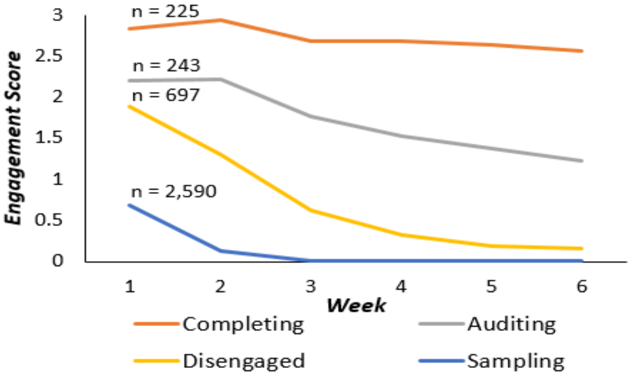


Figure 2. Results of Clustering Based on Learner Engagement Per Week.

3.2 Survey Data

Principal components analysis suggested that our scales were measuring three large components. Component 1 could be described as "mental health" and was comprised of PSS stress scores, PROMIS physical and mental health scores, PWB environmental mastery scores, SOSS lack of self-understanding scores, TIPI emotional stability scores, and FFMQ nonjudge and act with awareness scores. Component 2 could be described as "self and course beliefs" and was comprised of self-efficacy scores, beliefs about the effectiveness of yoga and meditation, and intentions to perform yoga and meditation. Component 3 could be described as "curiosity and openness" and was comprised of CEI embracing and stretching and TIPI openness and extraversion.

The discriminant function analysis was able to separate groups based on the results from the survey data. Function 1 had an eigenvalue of .023, accounted for 50% of the predicted variance, and had a canonical correlation of .15. Function 2 had an eigenvalue of .016, accounted for 35% of the predicted variance, and had a canonical correlation of .13. Function 3 had an eigenvalue of .007, accounted for 15% of predicted variance, and had a canonical correlation of .08. All functions taken together could significantly discriminate groups, Wilks λ = .96, χ^2 = 84.12, p < .001. With function 1 removed, functions 2 and 3 could significantly discriminate groups, Wilks λ = .98, χ^2 = 41.97, p = .025. Function 3 on its own could not discriminate groups, Wilks λ = .99, χ^2 = 12.84, p = .381. Group centroids for Functions 1 and 2 are presented in Figure 3.

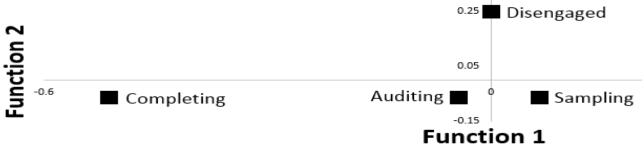
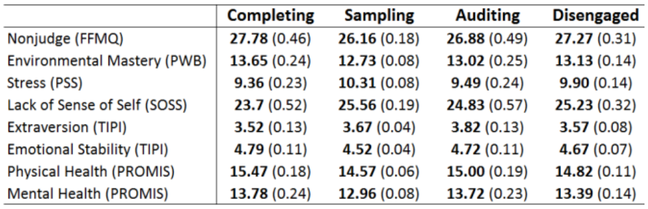


Figure 3. Group Centroids from Discriminant Function Analysis

Function 1 appears to separate Completing Learners from the rest, based on PWB environmental mastery scores, PSS stress scores, PROMIS mental and physical health scores, SOSS lack of self-understanding scores, FFMQ nonjudge scores, and TIPI emotional stability scores. Function 2 appears to separate Disengaged Learners from the rest, based on FFMQ nonjudge, TIPI extraversion, and TIPI emotional stability. Because of the small effect size, Function 3 was not interpreted. Mean values for the relevant scales are shown in Table 1.

Table 1. Survey Scores by Group, **M**(SE)



	Completing	Sampling	Auditing	Disengaged
Nonjudge (FFMQ)	**27.78** (0.46)	**26.16** (0.18)	**26.88** (0.49)	**27.27** (0.31)
Environmental Mastery (PWB)	**13.65** (0.24)	**12.73** (0.08)	**13.02** (0.25)	**13.13** (0.14)
Stress (PSS)	**9.36** (0.23)	**10.31** (0.08)	**9.49** (0.24)	**9.90** (0.14)
Lack of Sense of Self (SOSS)	**23.7** (0.52)	**25.56** (0.19)	**24.83** (0.57)	**25.23** (0.32)
Extraversion (TIPI)	**3.52** (0.13)	**3.67** (0.04)	**3.82** (0.13)	**3.57** (0.08)
Emotional Stability (TIPI)	**4.79** (0.11)	**4.52** (0.04)	**4.72** (0.11)	**4.67** (0.07)
Physical Health (PROMIS)	**15.47** (0.18)	**14.57** (0.06)	**15.00** (0.19)	**14.82** (0.11)
Mental Health (PROMIS)	**13.78** (0.24)	**12.96** (0.08)	**13.72** (0.23)	**13.39** (0.14)

4. DISCUSSION

These results represent two important findings. First, we showed that the learner clusters proposed by Kizilcec et al. were observable in our MOOC, therefore replicating previous findings. Second, we showed that these groups could be differentiated based on underlying psychological profiles. Primarily, we showed that Completing Learners were generally healthier psychologically than the rest of the learners, prior to participating in the MOOC. These findings suggest that underlying psychological factors may influence learner performance in MOOCs. Furthermore, the findings on perceived stress suggest that external environmental factors may also play a role in learner engagement patterns.

However, it is important to note that the effect sizes related to group discrimination in the present study were small. Therefore, although psychological factors may be able to discriminate these groups, other unobserved factors could be playing a much larger role. Future research is needed to further investigate group differences and increase our understanding of learner engagement. Additional research is also required to determine how to best apply this knowledge to ultimately improve learner success.

5. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Award Number 1546393

6. REFERENCES

[1] Kizilcec, R.F., Piech, C., & Schneider, E. (2013). *Deconstructing disengagement: Analyzing learner subpopulations in massive open online courses.* Learning Analytics and Knowledge Conference Proceedings. Leuven, Belguim.

[2] Hays, R.D., Bjorner, J.B., .Revicki, D.A., Spritzer, K.L., & Cella, D. (2009). *Development of physical and mental health summary scores from the patient-reported outcomes measurement information system (PROMIS) global items.* Quality of Life Research, 18, 873-880.

[3] Kashdan, T.B., Gallagher, M.W., Silvia, P.J., Winterstein, B.P., Breen, W.E., Terhar, D., & Steger, M.F. (2009). *The curiosity and exploration inventory-II: Development, factor structure, and psychometrics*. Journal of Research in Personality, 43, 987-998.

[4] Gross, J.J. & John, O.P. (2003). *Individual differences in two emotion regulation processes: Implications for affect, relationships, and well-being.* Journal of Personality and Social Psychology, 85, 348-362.

[5] Baer, R. A., Smith, G. T., Hopkins, J., Krietemeyer, J., & Toney, L. (2006). *Using self-report assessment methods to explore facets of mindfulness.* Assessment, 13, 27-45.

[6] Ryff, C.D., & Keyes, C.L.M. (1995). *The structure of psychological well-being revisited*. Journal of Personality and Social Psychology, 69(4), 719-727.

[7] Cohen, S., Kamarck, T., & Mermelstein, R. (1983). A global measure of perceived stress. Journal of health and Social Behavior, 24, 385-396.

[8] Flury, J.M. & Ickes, W. (2007). *Having a weak versus strong sense of self: The sense of self scale (SOSS).* Self and Identity, 9, 281-303.

[9] Gosling, S.D., Rentfrow, P.J., & Swann Jr., W.B. (2003). *A very brief measure of the big-five personality domains.* Journal of Research in Personality, 37, 504-528.

Using a Dynamic Bayesian Network to create a multidimensional longitudinal learner profile

Josine Verhagen
Kidaptive
101 Redwood Shores Parkway #130
Redwood City
+1 (669) 237-8320
jverhagen@kidaptive.com

Dylan Arena
Kidaptive
101 Redwood Shores Parkway #130
Redwood City
+1 (650) 345-5942
darena@kidaptive.com

ABSTRACT

We show some preliminary findings and propose a framework to use dynamic Bayesian networks with latent variables for combining information from different educational sources into a learner profile representing the strengths and weaknesses of a learner on multiple learning dimensions.

Keywords

Bayesian modeling, adaptive learning, Dynamic Bayesian Networks, psychometric models, game-based assessment

1. INTRODUCTION

Recent years saw an explosive increase in adaptive learning and intelligent tutoring systems. Most of these systems are focusing on adaptivity and feedback in one specific domain of study within a single educational product or environment. In this poster we will present some preliminary findings of a model we are working on that combines results from different educational environments into one comprehensive learner profile.

In our model, rather than using "knowledge components" we assume that interaction with educational content will result in relatively unidimensional and continuous latent ability estimates or scores like those resulting from a psychometric model like item response theory (IRT) [e.g. 1] or the Elo rating system [e.g. 2]. We also assume that the model to discover these scores (e.g., difficulty parameters for an IRT model) is known and fixed.

Our approach uses the resulting scores to track progress over time and combines scores from products related to similar learning dimensions in such a way that they represent progress on those learning dimensions. To represent the temporal component in this model we will use a dynamic Bayesian network [e.g. 3] with a first order Markov component.

2. MODEL

The full model (illustrated in Figure 1) consists of different parts: a *measurement model* part connecting item responses to a latent game scoreθusing the item difficulty b, a *factor model* part representing the factor loadings C of these game scores on the latent learning dimensions λ, and a *first order markov model* part representing the temporal component of the dynamic network. The first order Markov component can exist both on the level of the latent learning dimensions λ, as well as on the residual game score variance$w$ not explained by the latent learning dimensions. This can explain variation in game scores unrelated to the learning dimension, such as variation due to mastery of specific game mechanics. Equations 1-3 describe the structure of the model. The autoregressive weights for the latent learning dimensions are denoted by A, while the autoregressive weights for the score residuals are denoted by h. The residual latent score variance is denoted by e_t.

$$P(Y = 1 |\, \theta_t, b) \; = \frac{1}{1+e^{(\theta_t - b)}}(1)$$

$$\theta_t \; = C\Lambda_t \; + \; h\, w_{t-1} \; + \; w_t(2)$$

$$\Lambda_t \; = A\Lambda_{t-1} \; + \; e_t \;\; (3)$$

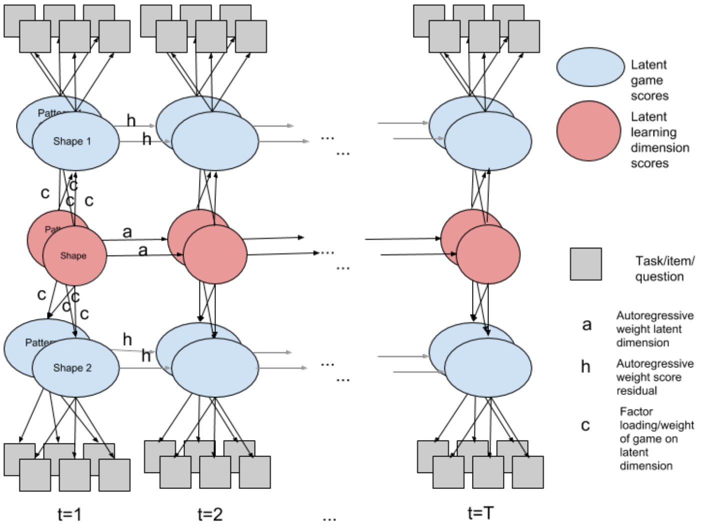

Patte
Shape 1
h
h
c
c
c
c
Patt
Shape
a
a
c
c
c
c
Patte
Shape 2
h
h
...
...
Latent game scores
Latent learning dimension scores
Task/item/ question
a
Autoregressive weight latent dimension
h
Autoregressive weight score residual
c
Factor loading/weight of game on latent dimension
t=1
t=2
...
t=T

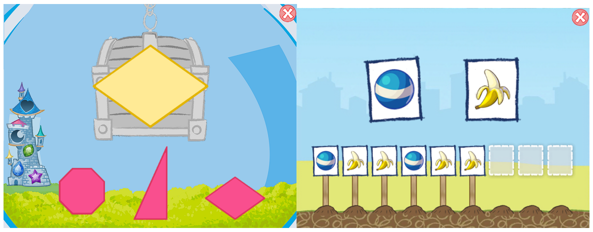
Figure 1. Dynamic Bayesian network structure

3. SHAPES AND PATTERNS

To show that the proposed model is identified and estimable, and to explore the usability of the resulting learning dimension scores, we use data from an app for preschoolers (2–5 years old) created to advance shape understanding and to teach pattern recognition and extension.

3.1 Data

The app contained two shape games (Figure 2) and two pattern games (Figure 3), along with many short educational video clips. In the two shape games, participants worked on shape identification and manipulation (translation, rotation, scaling, and composition). In the two pattern games, participants were shown a sequence of objects (such as ABAB, ABCABC, or ABBABB) and had to choose the correct object(s) to continue the pattern.



Shapes Game A Level 2 *Patterns Game D Level 7*
Task:Identify Advanced Shape *Task: Extend ABB(_)(_)(_)*

Figure 2: Screenshots of two games

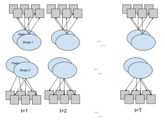
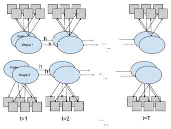
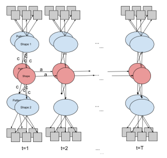
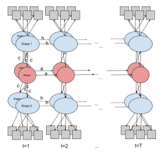
During six weeks, 91 participants in adaptive (44) and non-adaptive (47) conditions were asked to use the app. The six weeks were divided into 18 lessons, with each lesson lasting two or three days. During each lesson, participants were asked to play each of the four games at least once, after which they could replay games as often as they wished. Previously presented results [4] showed that scores estimated within the game showed increasing ability of participants in the shapes game, but not in the patterns game. Further investigation revealed that a large group of participants had trouble with even the easiest pattern questions, which would explain why they did not make much progress on this dimension.

3.2 Results

JAGS [5] was used to estimate a proof-of-concept model limited to four sources of information (the four games) and two latent dimensions (presumably shape and pattern knowledge).

The 18 lessons were used to represent separate time points *t*. To identify the latent learning dimension scores, in models 3 and 4 the loadings *c* of the first pattern and shape games on the latent dimensions were set to 1. Four different models with an increasing level of complexity were estimated, as described in Table 1.

Table 1. Description of models and DIC

Models	pD, DIC	
Model 1: Separate latent game score estimates for each game and time point	2333, 21202	
Model 2: Basic DBN with autoregressive component for each latent game score separately	1575, 20158	
Model 3: DBN with latent learning dimension scores + autoregressive component on the latent learning dimension scores only	1976, 20017	
Model 4: Full model, Model 3 + autoregressive component on the residuals of the game scores	1853, 19902	

Relative to a baseline model (Model 1) where latent game scores are estimated separately at each time point, the DIC improved when a temporal component was added (Model 2), and even more when latent learning dimensions (Model 3) and an autoregressive component on the residual latent games scores (Model 4) were added.

The estimated factor loadings and correlations showed that the shapes game scores (.7-1) but not the pattern game scores (.5-.4) were highly correlated with the first latent learning dimension scores. All games were highly correlated with the second estimated learning dimension score (.5-.9), although the patterns games somewhat higher. Inspection of the estimated latent scores revealed that the main trends of interest (e.g. the fact that regarding the patterns games, low ability learners were staying behind) were clearly visible in the estimated latent learning dimension scores

4. DISCUSSION

This work demonstrates that it is possible to estimate longitudinal multidimensional latent scores with a dynamic Bayesian network approach. The full model (Model 4) had improved fit compared to an several less complex models and led to, on first inspection, useable latent scores. This was a small example, however, and the scalability of this method is limited. In addition, we made some strong assumptions about the measurement models and time units.

We will continue to investigate more scalable methods of estimating dynamic Bayesian networks for multidimensional and longitudinal learner profiles based on data from different sources, and we will be running simulation studies to look at the limits of those methods.

On important future extension we plan to make is to anchor the latent learning dimension scores by independent standardized tests outside of the educational environment or game. This should lead to a more valid estimation and interpretation of the latent learning dimension scores.

5. REFERENCES

[1] Embretson, S. E., & Reise, S. P. (2013). *Item response theory*. Psychology Press.

[2] Klinkenberg, Sharon, Marthe Straatemeier, and Han LJ 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 (2011): 1813-1824.

[3] Murphy, K. P., & Russell, S. (2002). Dynamic bayesian networks: representation, inference and learning.

[4] Verhagen, J., Hatfield, D., Watson, J., Liu, S., & Arena, D. (2015). Shapes and Patterns of Adaptive Game-Based Learning: An Experiment. *Conference proceedings of the Games, Learning and Society Conference (GLS11),* 11, 248-254

[5] Plummer, M. (2012). JAGS: Just another Gibbs sampler. *Astrophysics Source Code Library*.

A Data-Mining Approach to Detecting Plagiarism in Online Exams

Sudipto Biswas, Edward F. Gehringer, Dipansha Gupta, Sanket Sahane, and Shriya Sharma
Department of Computer Science
North Carolina State University
+1 919-515-2066
{sbiswas4, efg, dgupta10, svshahan, ssharm25}@ncsu.edu

ABSTRACT

Online "open Web" exams present a more authentic assessment environment, but pose the risk of cheating via online messaging or access to shared web pages during the exam. This paper presents a data-mining approach to detecting cheating among students in online exams. Two techniques are explored: weirdness vectors and Levenshtein distances. The algorithms were tested on multiple-choice questions, fill-in-the blank questions, and essay questions. Both weirdness and Levenshtein approaches can provide a list of students who may have cheated, based on the similarity of their answers. Both approaches produce good results for objective (multiple-choice and fill-in-the-blank) tests, but only Levenshtein can detect suspicious similarities on essay tests.

Keywords

Online exams, plagiarism detection, collusion, MOOCs, Levenshtein distance

1. INTRODUCTION

Online exams are increasingly prevalent in MOOCs and distance education, and they are becoming more common in regular classrooms as well. Academic integrity is always a concern, and concern is heightened if students are allowed to surf the web during the exam, instead of being confined to a locked-down browser [1]. Current security approaches rely on watching the students, either with a live proctor or a webcam. But it is difficult, if not impossible, to catch all incidents of illicit communication between students, or students and outside parties. To address this issue, researchers started working on plagiarism detection in different languages since 1990. It was pioneered by a copy detection method in digital documents [6].

For the purposes of this paper, we define cheating as: sharing of answers between two or more students. Students may simply copy or modify answers of peers to pass off as their own. They may, for example, take a peer's answer and make small changes by either adding, deleting or substituting particular characters to that answer and submitting it.

Data mining can uncover potential violations. We hypothesize that two students having the same wrong answer to the same questions (or question parts) is indicative of possible plagiarism or illicit communication between them. In order to detect this, we make use of two potential approaches: a "weirdness" vector, and weighted Levenshtein distance. We apply these two algorithms to two different types of exams: exams with mostly multiple-choice questions and exams with mostly essay questions.

2. RELATED WORK

Plagiarism detection can be formalized as a problem of computing a similarity between documents [2]. Some researchers have focused on String Similarity Metric [7]. Others focus their work on vector distance calculations [8]. Another approach is using statistics of word occurrences such as the bag-of-words model [3]. One might also use patterns of word occurrences, such as the edit distance and its weighted and local versions [4]. There already exist plagiarism-detection methods based on modifications to the Levenshtein distance algorithm [5] but they are complex. We seek a simpler approach.

3. ALGORITHMS

3.1 Weirdness vector

Let us assume that two exams are suspiciously similar if they have very unusual,, or "weird" answers for the same questions or question parts. We define a weird answer as an answer to a particular part of a particular question that is very unusual among all the answers submitted for that question part. In other words, the *term frequency* of this answer is low, among all the answers submitted to this question part.

The basic idea is that if two students have weird answers in the same places on the exam, and moreover, they are the *same* weird answers, then their submitted answers should be examined carefully for evidence of plagiarism.

For each student, we can create a "weirdness vector," which is formed from the weirdness values for that student's answers to each question on the test. Weird answers have a low frequency of occurrence, i.e., a frequency near 0. However, we compare weirdness vectors using cosine similarity, which requires nonzero values. So instead of using the frequency *f* directly, we form the weirdness vector from values $1-f$ for each of the student's answers.

Cosine similarity is defined as

$$\text{Cosine similarity} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

where x and y are the components of the two vectors X and Y say. Cosine similarity between two non-zero vectors is a measure of their similarity. It measures the cosine of the angles between the two vectors.

Zeros in either of two vectors between which cosine similarity is measured do not contribute or influence the similarity value; only non-zero values are of importance. Thus, high-frequency answers contribute little to cosine similarity, and cause the similarity measure to depend heavily on the answers that are really unusual.

The program reports the top matches, and the instructor can look at the exams of these students to verify whether there are suspicious similarities. As in programs that detect plagiarism in programming assignments, a high similarity may not be indicative of unauthorized collaboration. It really depends on the type of questions. For multiple-choice questions, where only a few answers are possible, there is much more of a chance of answers matching by accident than for fill-in-the-blank questions, where student answers can consist of arbitrary strings of text. Consequently, a high cosine similarity for a multiple-choice test of k questions is much less suspicious than it would be for k fill-in-the-blank questions, or for a multiple-choice test of $m >> k$ questions.

3.1.1. Weirdness-based plagiarism algorithm

The following is a step-by-step description of the algorithm that uses Weirdness Vector to detect similar answers:

A weirdness score w is generated for each student's answer to each question part, by a term-frequency calculation on the set of answers to this question part provided by the whole class.

Let f be the proportion (frequency) of students having a given answer a to a question part q. Then w is set to $1-f$. Thus, an answer that is "weird" will have a higher w value than an answer that is common. The weirdness score is rescaled to range between 0−1 as explained above. The pairs of students with cosine similarity higher than a threshold value t are generated and displayed to the instructor.

Among all question types, this approach works best for fill-in-the-blank questions. These questions allow a much greater diversity of answers than multiple-choice, checkbox, or matching questions. Short-answer, or essay questions have an answer space that is much larger still, but here, cosmetic differences in wording, or even whitespace, can prevent a match from being detected.

3.1.2. Pseudo-code for weirdness algorithm

Here is the pseudo-code for the weirdness algorithm.

```
1.  for the ith row in A,
        a.  for the jth column in A,
                i.   $f_{i,j}$ = term frequency of $A_{i,j}$
                ii.  $w_{i,j} = 1 - f_{i,j}$
2.  for the ith row in w,
        a.  for the jth row in w,
                i.   $d = cosine\_similarity(w_i, w_j)$
                ii.  $scores.append(i, j, d)$
3.  display scores with $d > t$
```

In the above algorithm,

A = matrix of answers by students where $A_{i,j}$ represents the answer given by the ith student to the jth question.
w = matrix of weirdness of each answer where $w_{i,j}$ represents weirdness of the ith student's jth answer.

3.2 Levenshtein distance

One limitation of "weirdness" is that it looks for which answers have low frequency, without taking into account answers that are similar to those answers. In the real world, plagiarized answers might differ in wording and whitespace, and we would like to detect them even when they are cosmetically different. Toward that end, we implemented another algorithm that makes use of Levenshtein distance. Levenshtein distance is the minimum edit distance between two strings. It is defined as the number of characters to be changed in order to change one string into the other.

For every pair of students, we consider what the Levenshtein distance is between each of their corresponding answers. Assuming this distance is d, the similarity becomes $1-d$. For example, consider two students whose answers are "hello" and "hello world". In this case, 6 out of 11 characters in the second string are new; thus the distance $d = 6/11 \cong 0.6$, so the similarity = 0.4. And indeed, the strings are approximately 40% similar. A higher similarity means that fewer edits were required to convert one answer into the other. This helps detect the case where a student copies an answer from another student and edits it to make it look different. Thus Levenshtein distance serves as a good metric to detect a possibility of plagiarism.

Just as for weirdness, we can collect Levenshtein distances into a vector. This vector holds values that measure similarities between corresponding answers of those 2 students. We use the *median* of these values as a proxy for the overall distribution, and let this represent our estimate of the likelihood that the students have cheated. This prevents the estimate from being unduly affected by a few large or small Levenshtein distances.

Consider a 5-question exam where the similarity vector for 2 students is [0.2, 0.4, 0.7, 0.8, 0.9]. The median of values in this vector is 0.7. The probability of plagiarism seems like it might be high (of course, we would have to look at the answers themselves to make a judgment). Now consider the vector [0.2, 0.2, 0.3, 0.4, 0.5] with median 0.3. Most values in the second vector lie near 0.3.

The lack of similarity in the corresponding answers makes it appear less likely that this pair of students have cheated.

3.2.1. Overview of Levenshtein distance algorithm

The following is a step-by-step description of the algorithm that makes use of Levenshtein distance:

- For every pair of students, a vector *v* is calculated, consisting of the Levenshtein similarities for each question for this pair of students.
- For each vector, calculate the median value.
- The pairs of students with the median higher than a threshold value *t* are displayed to the instructor.

3.2.2. Pseudo-code for Levenshtein distance algorithm

The following is the pseudo for plagiarism detection using Levenshtein distance algorithm:

```
● for the $i$th row in $A$
    ○ for the $j$th row in $A$
        ■ If $i = j$, skip
        ■ for the $k$th column in $A$
            ● $vector_k = levenshtein(A_{i,k}, A_{j,k})$
    ○ $similarity_{i,j} = median(vector)$
● display similarity values greater than threshold
```

In the above algorithm,

A = matrix of answers by students where $A_{i,j}$ represents the answer given by the *i*th student to the *j*th question.
similarity = matrix of similarity where $similarity_{i,j}$ represents similarity between the *i*th student and the *j*th student.

3.2.3. Levenshtein distance over other distance types

Many cases of plagiarism originate from a student copying an answer and then changing a few words or characters to make it look like his/her own. So Levenshtein seems an appropriate metric for comparing free form text answers.

Instead of Levenshtein distance, we could use the Hamming distance. This distance denotes the number of places where string S_1 is different from S_2. This kind of distance seems to be a poorer metric for our purposes than Levenshtein distance because Hamming distance takes into account only the edits, i.e., replacements of characters, whereas Levenshtein takes into account insertions and deletions as well as edits.

4. EXPERIMENTAL TUNING OF VARIABLES

Several variables are used in both algorithms can be varied to get better results. Section 4.1. and 4.2 explain them and how the variation may have an effect on the overall results.

4.1 Threshold value in weirdness

We make use of a threshold value t while displaying the results. This value denotes the similarity score beyond which the students are considered potentially to have cheated. For some exams this threshold value may be very high, e.g., multiple-choice exams. For such exams, only a limited number of answers are possible, and thus high similarities are very possible by chance. Therefore in this case, setting a very high threshold avoids identifying pairs of students who did not really collude. Similarly for an exam where essay questions dominate, even a similarity of 60% may provoke suspicion of plagiarism. In this scenario, a lower threshold may work better. Again, these are speculations and a value that works well for an exam can best be found by experimentation.

4.2 Median in Levenshtein distance

It is also possible to experiment with using metrics other than the vector median to represent the possibility of plagiarism. For example, we might use the 3rd quartile of values in the vector as a measure of how likely the students were to have cheated. The 3rd quartile would work well if students who cheated plagiarized only ¼ of the answers on the exam.

5. IMPLEMENTATION AND RESULTS

The two algorithms were tested on multiple choice (MCQ) exam taken by about 68 students, and a subjective essay exam taken by 105 students. For each of the tests we then plotted the weirdness score and the weighted Levenshtein scores of each student pair and the graphs have been shown below.

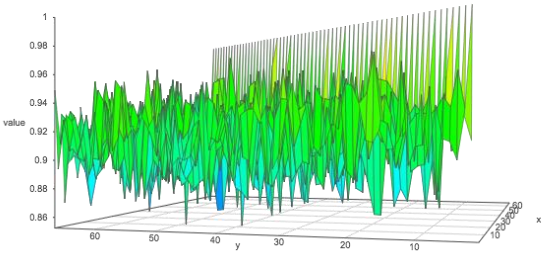


Figure 1 Similarity plot for weirdness on a MCQ exam

The weirdness algorithm when tested on multiple choice type questions returned similarities which mostly lie between 0.8 and 1, as shown in Figure 1.

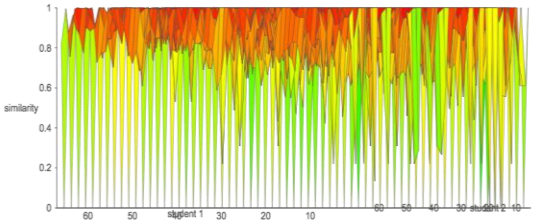


Figure 2 Similarity plot for Levenshtein on a MCQ exam

When the Levenshtein approach was tested on the same file it returned a good spread of values with similarities lying either very close to 0 to or very close to 1, as shown in Figure 2.

This is because the MCQ answers are usually a choice between one of four or five characters. So either students chose the same answers (same characters), which resulted in a similarity score of 1 or they didn't, which resulted in a score of 0.

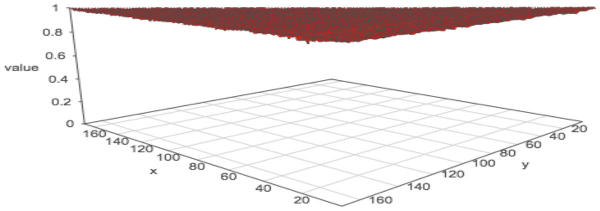


Figure 3. Similarity plot for weirdness on an essay exam

We then ran the two algorithms on essay type questions. Plotting the similarity scores from the Weirdness algorithm we see that most of the answers lie at at around 1 as shown in Figure 3.

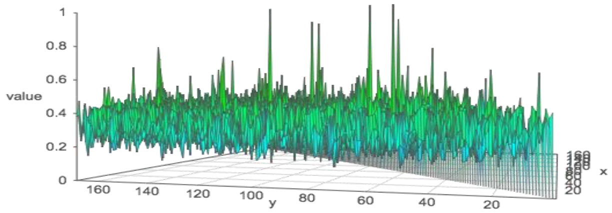


Figure 4 Similarity plot for Levenshtein on an essay exam

The plot of the similarity scores from the Levenshtein algorithm when plotted, demonstrates a wide range of similarity scores, as shown in Figure 4. Ostensibly this is because essay questions admit a wider range of responses, depending on students' knowledge and writing style. A high similarity score is thus more suspicious than it would be on a more objective exam.

6. SUMMARY

We observed in conclusion that Levenshtein works better than Weirdness in case of essay type exams. For MCQs results of both the algorithms are comparable. In addition to this, adding a metric to incorporate the correct answers may prove helpful. We have devised a method to do so but we are still experimenting with the results to see what effect it may have on the current results.

7. REFERENCES

[1] Edward F. Gehringer and Barry W. Peddycord, "Experience with online and open-Web exams," *Journal of Instructional Research* 2, 2013, pp. 10–19.

[2] R. Lukashenko, V. Graudina, and J. Grundspenkis. Computer-based plagiarism detection methods and tools: An overview. In Proceedings of the *2007 International Conference on Computer Systems and Technologies*, pages 1-6. ACM, 2007.

[3] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[4] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences, *J. Mol. Biol.*, 147:195-197, 1981.

[5] Z. Su, B.-R. Ahn, K.-Y. Eom, M.-K. Kang, J.-P. Kim, and M.-K. Kim. Plagiarism detection using the Levenshtein distance and Smith-Waterman algorithm. *In Innovative Computing Information and Control*, page 569, 2008.

[6] S. Brin, J. Davis, H.Garcia-Molina, Copy detection mechanisms for digital documents, in: *ACM SIGMOD Record*, Vol. 24, ACM, 1995, pp. 398-409.24

[7] M. Elhadi, A.Al-Tobi, Duplicate detection in documents and web pages using improved longest common subsequence and documents syntactical structures, in: *Computer Sciences and Convergence Information Technology, 2009.* ICCIT' 09. Fourth International Conference on, IEEE,2009, pp.679--684.

[8] H. Zhang, T. W. Chow, A coarse-to-ne framework to efficiently thwart plagiarism, *Pattern Recognition* 44 (2) (2011) 471-487.

Development of an Educational Dashboard for the Integration of German State Universities' Data

Alexander Askinadze and Stefan Conrad
Institute of Computer Science
Heinrich Heine University Düsseldorf
Universitätsstr. 1
D-40225 Düsseldorf, Germany
{askinadze, conrad}@cs.uni-duesseldorf.de

ABSTRACT

German state universities often only have little data about their students. Existing data includes study history data such as grades, module names, number of attempts, and dates. This data can be used to extract interesting information, although it is often not used. It is stored in various databases and systems in the universities, so that each university has to develop its own analysis tools. We have developed a tool that allows various universities to easily import the data and retrieve first visualizations of it.

Keywords

educational dashboard, student data integration, educational data visualization

1. INTRODUCTION

German state universities often do not offer any e-learning in addition to the usual teaching materials, so there is usually no data from interactions with an e-learning system. Only the study course data such as grade, module name, number of attempts, and date are available. Even if there are a few features, this data could be analyzed. Since state universities often have no educational dashboards or evaluation systems, the existing data is not used. The data is available in different database systems and formats, so each university would have to develop its own system to generate knowledge from its own data.

To solve this problem, we developed a dashboard which should be able to integrate data from different universities. The courses are usually organized in modules, which can be constructed in a hierarchical structure so that, for example, the module "Mathematics" is considered as passed if the submodules "Calculus I" and "Linear Algebra I" are passed. The system must, therefore, be able to integrate various hierarchical structures.

Once the data is integrated, it can be visualized and data mining procedures can be applied to it. The development of the visualization and data mining procedures can be done in a central place and does not have to be redeveloped at every university. The individual universities should only have to synchronize their data with our dashboard.

2. METHOD

To integrate data from different universities, we propose a simplified data model. Four of the required tables are shown in Figure 1. The *student* table contains all necessary students' data. The more attributes, the more information can be analyzed. However, the features *Completed*, *EnrollmentDate*, and *ExmatriculationDate* at least should be specified. *Completed* denotes students who have successfully completed their studies. Students who have an *ExmatriculationDate* and at the same time have the value *false* in the *Completed* field can be regarded as students who have dropped out of their studies.

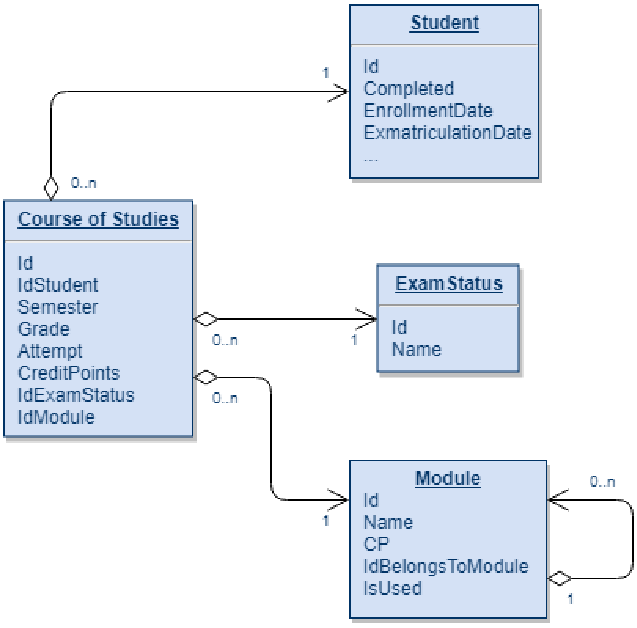


Figure 1: Simplified version of the model

To import the data, the universities only need to export two tables: students and *student achievements*. Table 1 gives two examples of student data. The student with Id 1 graduated successfully and the student with Id 2 is still studying.

Table 1: Student Data

Id	Completed	EnrollmentDate	ExmatricualtionDate
1	True	2009/10/01	2012/09/30
2	False	2009/10/01	null

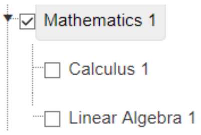
Table 2 shows the achievements of two examples in which the student with Id 1 passed two modules X and Y at the first attempt and gained a total of 20 credit points (CP) in the 1st semester.

Table 2: Student Achievements (Course of Studies)

Id Stud	Semester	Attempt	Exam status	Module	Parent Module	CP
1	1	1	Passed	X	Z	10
1	1	1	Passed	Y	Z	10

The hierarchy of the modules can be extracted from this table based on the parent relation of the modules.

Each university is able to freely select the module hierarchy level that should be used for the analysis. We suggest that this should be chosen with a clickable treeview. Figure 2 illustrates how the module *Mathematics* containing the overall information of its submodules is selected.



☑ Mathematics 1
□ Calculus 1
□ Linear Algebra 1

Figure 2: Selecting the required module for the data analysis

By combining the data from Tables 1 and 2, the model specified in Figure 1 can be filled in and the first analyses automatically created. Below are several practical visualizations that can be created from such data.

The heat map in Figure 3 shows students who dropped out their studies. Each row shows a student and each column shows how many exam attempts they had each semester. For example, we see that students who drop out in the first semester usually do not have more than 2 exam attempts.

To find out which combinations of exams are passed together, Venn diagrams can be used. Our dashboard generates Venn diagrams for the combination of selected drop out semesters and selected exams. The Venn diagram in Figure 4 visualizes which exams are passed by students who drop out in the 1st semester. For visualization, the four modules Calculus I, Linear Algebra I, Technical Computer Science and Operating Systems were selected, with Operating Systems not being part of the syllabus in the 1st semester. From Figure 4 we can see that students who pass Calculus I are also able to pass the other planned exams. For further information, one could compare this with a Venn diagram that shows the attempts made.

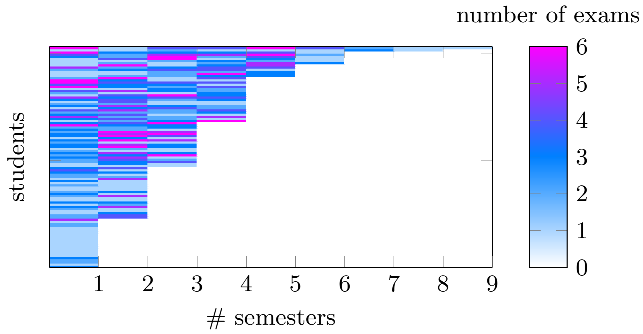


Figure 3: Heatmap visualization of study progress of students before they have canceled their studies. The number of exam registrations per semester is shown with colors in each row.

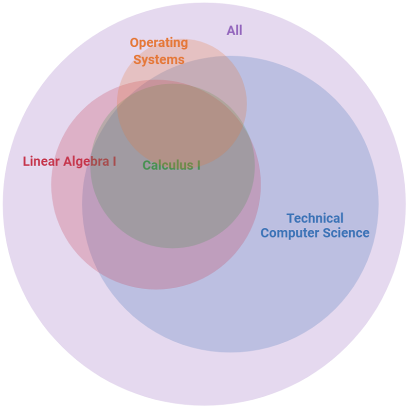


Figure 4: Exams passed in the 1st semester by students who drop out after the 1st semester

Figure 5 shows a violin plot of graduates and how many CP they have earned per semester. This visualization shows us that successful students with a few exceptions earned approx. 10-40 CP per semester. With the violin plots, averages and distributions for different features can be clearly visualized.

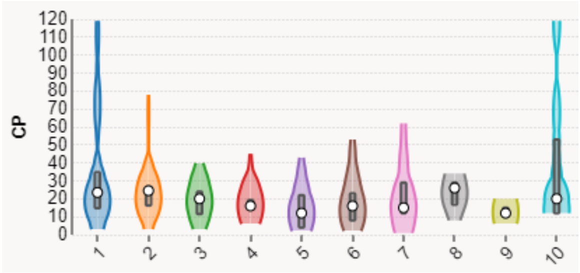


Figure 5: CP per semester of graduated students

3. CONCLUSION AND FUTURE WORKS

We have presented a first version of the dashboard that is able to import data from various state universities and to visualize the data. In the future, we will implement additional visualizations as well as add predictive models.

How are Students Struggling in Programming? Understanding Learning Processes from Multiple Learning Logs

Yuta Taniguchi
Faculty of Information Science
and Electrical Engineering,
Kyushu University
Fukuoka 819-0395, Japan
taniguchi@ait.kyushu-u.ac.jp

Fumiya Okubo
Faculty of Business
Administration,
Takachiho University
Tokyo 168-8508, Japan
fokubo@takachiho.ac.jp

Atsushi Shimada
Faculty of Information Science
and Electrical Engineering,
Kyushu University
Fukuoka 819-0395, Japan
atsushi@ait.kyushu-u.ac.jp

Shin'ichi Konomi
Faculty of Arts and Science,
Kyushu University
Fukuoka 819-0395, Japan
konomi@artsci.kyushu-u.ac.jp

ABSTRACT

This study discuss how students resolve compilation errors with textbooks in C programming exercises. A student's understanding of the programming language is strongly connected with compilation results. Resolving compilation errors requires one to check source code carefully and understand the language better. Reading textbooks is a typical learning activity of students in resolving errors as well as asking teachers. Therefore, learning processes in a programming exercise course could be understood by combining students' compilation logs and reading logs of e-textbooks. In this paper, we present preliminary results of analysis on students' struggling to resolve errors. Using a dataset collected during a semester in our university, we discuss the universality of errors in terms of students and exercise questions. Furthermore, we reveal the positive and negative impact of reading e-textbooks on error resolutions on a per-page basis.

Keywords

programming exercise, learning process, error resolution

1. INTRODUCTION

Programming techniques are getting more and more attention recent years, and are being introduced into educational curriculum in primary and secondary education as well as higher education. The C programming language is one of the most important and popular programming language widely used in industries over the past decades. However, there are many obstacles for students in learning the programming language, and thus how to understand and support their learning is an important question.

On the one hand, we need to support students to fix errors in their source code. It is said nearly half of the time and effort are spent in debugging during the development of a program [3]. Park et al. reported that common syntax errors tend to remain in source code for a long time [4]. It requires too much effort for a student to identify and fix such errors in learning.

On the other hand, we have to grasp how students struggle to resolve errors in source code. We can consider that many types of mistakes in source code reflect a student's understanding at some point of his or her learning process. Tracking the resolution of compilation errors, we could understand the learning processes of students better.

To this end, many work analyzed errors in programming courses. Fu et al. [1] have proposed a web-based system that helps teachers to support student during class by providing real-time dashboard to grasp students' learning situations. Their system is helpful to overview the current situation of a single class at a glance. However, the information on the dashboard is somewhat superficial and based on short-period data, and how they struggle to resolve errors is not discussed.

Helminen et al. [2] addressed the process in which students struggle to resolve errors. However, in their study, only limited activities of selecting, ordering, and indenting code fragments are analyzed, and activities such as referring external learning materials are not considered. For understanding students' learning processes, it is significant to know how students search learning resources for necessary information and acquire knowledges. Nevertheless, only a limited number of studies focused on students' try-and-error and knowledge acquisition in learning processes of programming languages.

In this paper, we analyze how students struggle to resolve compilation errors with course materials in a programming exercise course. Toward this end, we employ both compilation logs and page view logs of e-textbooks, and characterize compilation errors in relation to exercise questions and individual students. Furthermore, our preliminary result shows the positive or negative contribution of every page of course materials in resolution of a particular error.

2. METHOD

2.1 Compilation and e-Book Operation Logs

We focus on the data obtained from the multiple classes of the C programming course of our university offered in the first semester 2017. The course is manly for freshmen, and includes lectures and coding exercises. There are about 20 classes for the course in a semester, and almost all of the courses are taught by different teachers. We have a set of standard course materials and usually they use it in their teaching, but it is not enforced.

In exercise, we use the compiler "gcc", from the GNU compiler collection, on a remote Linux server. The compiler program is modified from the original version so that it can record students' learning logs. More precisely, when it is executed, it saves given commandline arguments, the contents of given source files, and the output of the compiler as well as the time and user of the invocation. Since a commandline and source code are available as logs, we can reproduce what a student tried and what he or she obtained as a result.

In most cases, the compiler's output is produced only when there are some problems. The majority of problems are in source code, which result in compilation errors. The others are caused by errors outside source code, such as inadequate arguments and wrong filenames, and they are not recognized as compilation errors. We ignore the latter type of errors in this study since we are not interested in the learning process of compilation itself but in that of a programming language.

We also utilize students' activity data of reading course materials. Our materials are provided on our own e-book system. Students read those materials on the web, and their operations on the system are collected immediately as events. There are variety of operation types including page flipping, full text searching, bookmarking, and so on.

Combined with compilation logs, these event logs tell us how students learned during exercise. For example, after a compilation failure, some students just repeat compilation without necessary modification of source code, and some other students go back to course materials and try to find the key to solve the problem. It might be also possible to evaluate the ability of students. If a student can quickly rewrite source code without looking any materials when a compilation failed, we can consider he or she is well experienced. We should care a student who read many pages of course materials and still failing to compile their source code.

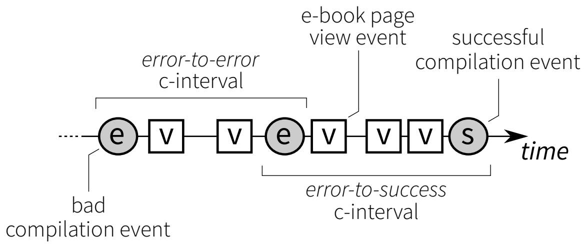

e-book page view event
successful compilation event
error-to-error c-interval
e
v
v
e
v
v
v
s
time
bad compilation event
error-to-success c-interval

Figure 1: An example of a session and c-intervals.

```
4_a-1.c: In function ‘main’:
4_a-1.c:9: error: conflicting types for ‘X’
4_a-1.c:5: note: previous declaration of ‘X’ was here
```

Figure 2: An example output of compilation error. The first line describes a context where the following error of the second line happened. The final line represents an error, but a note. Only a single error is included in this example output.

2.2 Timeline

We call a sequence of events a *timeline.* A timeline is composed of compilation events and e-book system's page view events, and represents a students' activities in a class. A timeline can be divided into shorter parts by exercise problems that a student was working on at a point of time. In exercise, students are given several problems to solve and instructed to save source code for the problems in separate files with specified filenames. Since a filename and a exercise problem is connected, we can identify a corresponding task from a compilation event log and split timelines into some parts. We call such a part *session* in this paper.

Furthermore, we introduce the concept of *c-interval.* A c-interval is a part of a session which starts and ends with compilation events, and includes only non-compilation events between them. This is a basic unit considered in our analysis because we are interested in what a student does after a compilations and how such activities affect the succeeding compilation.

Seeing if compilation events of a c-intervals are successful or not, we can classify c-intervals into four types: *error-to-error*, *error-to-success*, *success-to-error*, and *success-to-success*. Figure 1 shows an example of a session. In this example, a session consists of eight events; three are compilation events and five are e-book system's page view events. Two types of c-interval, *error-to-error* and *error-to-success* ones, are also shown.

2.3 Compilation Errors

First of all, we normalize the language of error messages. Some of the error messages could be recorded in a non-English language of a student's preference. In our dataset, most of error messages are written in English, but some Japanese or Chinese words are also included. Therefore, we translate such non-English portions of messages into English by a dictionary based method. Please note that the dictionary is currently incomplete and it affects the results shown later in this paper.

```
c_b-1.c: In function ‘main’:
c_b-1.c:8: error: expected ‘;’ before ‘}’ token
c_b-1.c:9: error: ‘else’ without a previous ‘if’
c_b-1.c:9: error: expected ‘;’ before ‘}’ token
c_b-1.c:10: error: ‘else’ without a previous ‘if’
c_b-1.c:10: error: expected ‘;’ before ‘}’ token
```

Figure 3: An example of compilation output with multiple errors. In this example five error messages are included led by a message describing a context in which those errors happened.

```
{param1}: error: stray ‘{param2}’ in program
{param1}: warning: null character(s) ignored
{param1}: error: expected ‘{param2}’ before ‘{param3}’ token
{param1}: error: expected ‘{param2}’ before ‘{param3}’
{param1}: error: expected expression before ‘{param2}’ token
{param1}: error: expected expression before ‘{param2}’
{param1}: error: too few arguments to function ‘{param2}’
{param1}: error: stray ’{param2}’ in program
{param1}: error: conflicting types for ‘{param2}’
{param1}: error: invalid suffix "{param2}" on integer constant
```

Figure 4: Example templates of error messages obtained from our dataset.

We identify every error message in compiler output with our own parser. We developed a parser program that automatically identify error messages in compiler output based on a heuristic algorithm. Figure 2 shows an example output of compilation error. The first line describes a context where the following error of the second line happened. The final line represents an error, but a note. Consequently, our program identifies only a single error in this example. Figure 3 shows another example output of compilation error with multiple errors. In this example 11 error messages are identified by the parser which is led by a message describing a context in which those errors happened.

Many of found error messages share the same underlying structures, for example:

```
4_a-1.c:9: error: conflicting types for ‘X’
8_b-2.c:21: error: conflicting types for ‘count’
```

Such a structure can be represented as a single template like

```
{p1}:{p2}: error: conflicting types for ‘{p3}’
```

where `{p1}`, `{p2}`, and `{p3}` are placeholders of the template. We believe such a template represents the essentials of errors better than raw messages. Hence we identify a template text with an error in later analysis.

We investigated actual error messages and found out heuristic rules to obtain a template from a message. Based on the rules, we extracted all the templates from our dataset. There are 52,384 different error messages in our normalized dataset, and they were greatly reduced into 247 message templates. Figure 4 shows a set of examples from them.

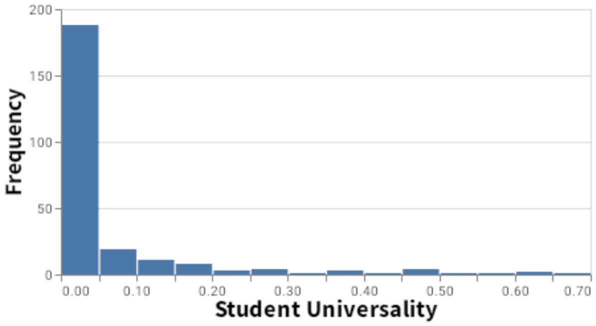


Figure 5: Histogram of student universality of errors. The horizontal axis indicates student universality of errors, and the vertical axis shows the number of errors.

2.4 Analysis

We analyze the universality of errors to know the diversity of students' struggling. Since every student have different understanding, it is expected that there are only a few common errors and many student-specific errors. We also consider the topics of exercise questions influence the tendency of compilation errors. Therefore, we consider two kinds of universality: one based on student and the other based on questions.

Given an error, the former universality can be quantified as the ratio of students out of all students who encountered the error. The latter can be similarly computed as the ratio of questions among all exercise questions where students encountered the error. We call these ratios *student universality* and *question universality* of an error, respectively.

We also analyze how reading material pages impacted on error resolution in students' struggling. To this end, we focus on *error-to-error* and *error-to-success* c-intervals. We assume that all pages viewed during these types of c-intervals were for fixing compilation errors, especially errors found in the earlier compilation event of c-intervals. In the case of *error-to-success*, we consider pages contributed positively in error resolution; contrastingly, we consider pages worked negatively in the case of *error-to-error*.

The contribution of every page are measured for each error as follows. We count how many times a page p_i contributed positively or negatively for the resolution of an error e_j. Given p_i and e_j, let $C^{i,j}_{\text{pos}}$ and $C^{i,j}_{\text{neg}}$ be the numbers of positive or negative cases, respectively. The contribution of p_i for e_j is defined as follows:

$$Contribution(p_i, e_j) = \frac{C^{i,j}_{\text{pos}} - C^{i,j}_{\text{neg}}}{C^{i,j}_{\text{pos}} + C^{i,j}_{\text{neg}}}$$

With this formulation, contributions are represented by values in $[-1, 1]$. A positive value indicates more positive contributions than negative ones, and vice versa.

3. EXPERIMENT

Figure 5 shows the distribution of the student universality of errors as a histogram. The horizontal axis indicates the student universality, and the vertical axis shows the frequency

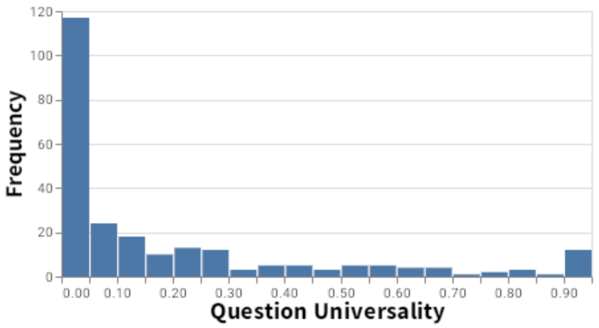


Figure 6: Histogram of the question universality of errors. The horizontal axis indicates the question universality of errors, and the vertical axis shows the frequency of errors.

of errors. From the figure, we can say that there is a small number of common errors, while most of the errors occur to a limited number of students. To be precise, about the three quarters of the errors were encountered by at most five percent of students, and about the half of the errors were encountered by at most one percent of students. Moreover, the common errors that a majority of students encountered were limited to only five particular types.

Figure 6 shows the distribution of the question universality of errors as a histogram. The horizontal axis indicates the universality, and the vertical axis shows the frequency of errors. The figure shows there are more than ten errors that are not problem-specific. It seems that most errors are associated with a few exercise problems. However, compared to that of student universality, there are more errors which occurs for more than half of questions.

Figure 7 is the heatmap that shows contributions of material pages to error resolutions. Each row corresponds to an error, and each column represents a page of a material. Pages are sorted by the order of material usage and then by page numbers. In the heatmap, a cell with positive contribution value is colored reddish. In the reverse case, a cell is colored bluish.

From the figure we can observe positive and negative cases are clearly separated in fairly many cases. This fact suggests that pages does matter for resolving compilation errors. It also seems that students tend to read many useless pages when they struggled to fix errors. Consequently, it may be helpful for teachers to teach student how to understand error messages and find effective material pages.

4. CONCLUSIONS

In this study, we investigated how students struggle to resolve compilation errors with textbooks during exercises, and we employed compilation logs and browsing history of e-textbooks to this end. As the preliminary results, we found that most of the errors are student- and question-specific, and errors universally observable are quite limited. Reading course materials seems to be helpful for error resolution though it is highly dependent on a type of errors. Hence, we conclude that students have different situations individually and they have to find helpful material pages to resolve particular errors. This suggests the need of personalized analysis and support in programming education. The limitation of the work includes the lack of student-wise and statistical analyses. We are going to do more personal analysis with more data through several semesters as future work.

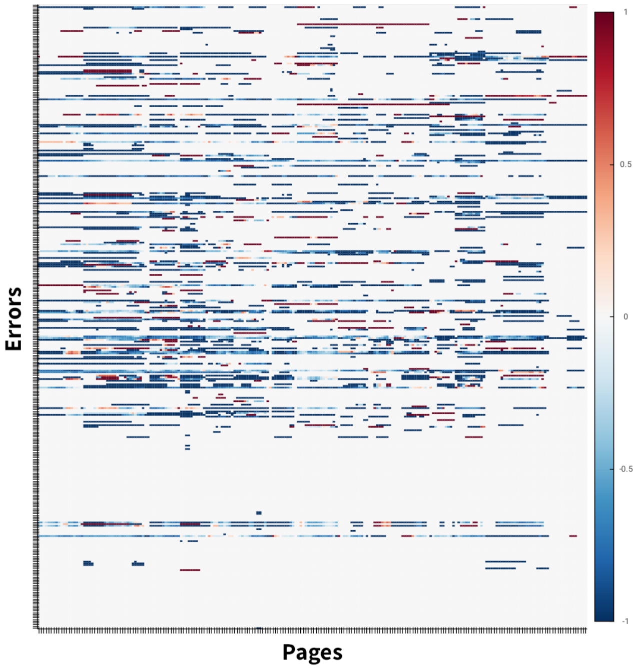


Figure 7: Heatmap showing contribution of material pages to error resolutions. Each row corresponds to an error, and each column represents a page of a material. Reddish color represents positive contribution values, and bluish color represents negative values.

5. ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number JP17K12804.

6. REFERENCES

[1] X. Fu, A. Shimada, H. Ogata, Y. Taniguchi, and D. Suehiro. Real-time learning analytics for c programming language courses. In *Proceedings of the Seventh International Learning Analytics &#38; Knowledge Conference*, LAK '17, pages 280–288, New York, NY, USA, 2017. ACM.

[2] J. Helminen, P. Ihantola, V. Karavirta, and L. Malmi. How do students solve parsons programming problems?: An analysis of interaction traces. In *Proceedings of the Ninth Annual International Conference on International Computing Education Research*, ICER '12, pages 119–126, New York, NY, USA, 2012. ACM.

[3] G. J. Myers. A controlled experiment in program testing and code walkthroughs/inspections. *Commun. ACM*, 21(9):760–768, Sept. 1978.

[4] T. H. Park, B. Dorn, and A. Forte. An analysis of html and css syntax errors in a web development course. *Trans. Comput. Educ.*, 15(1):4:1–4:21, Mar. 2015.

Automatic Learning Pathway Construction in Connected Learning Environments

Joshua Ladd
University of Colorado Boulder, Department of Computer Science
Joshua.Ladd@colorado.edu

Katie Van Horne
University of Colorado Boulder, School of Education
Katie.VanHorne@colorado.edu

Ahmed Mohamed Fahmy Yousef
University of Colorado Boulder, Institute of Cognitive Science
amf00@fayoum.edu.eg

Tamara Sumner
University of Colorado Boulder, Institute of Cognitive Science
sumner@colorado.edu

ABSTRACT

Connected learning is a model that aims to enable youth to explore their interests across settings. However, finding learning opportunities that are aligned to their interests can be challenging. Creating structured pathways by hand is also too time consuming to be done at scale. This poster introduces a scalable computational approach for automatically constructing interest-driven pathways through a network of informal learning opportunities using graph traversal and natural language processing.

1. INTRODUCTION

Connected learning is a model for helping students discover new interests and pursue interest-related learning across different settings [1]. The abundance of online and face-to-face learning opportunities can allow students to follow their interest down a multitude of pathways, but the sheer variety and amount of opportunities can be daunting to choose from. For students to effectively deepen their interests, it is important to help them determine what they want to explore.

Chicago Cities of Learning (CCOL) is an online environment that provides youth in Chicago with access to informal learning opportunities across the city by enabling organizations to list and advertise their programs and creating institutional linkages between organizations. While CCOL provides youth with entry points to explore their own interests, it does not have an organized set of pathways across all program areas to help youth deepen their experiences in their interest-related pursuits. There is limited evidence from the site's badging system that youth created their own pathways in pursuit of these badges, rather most followed suggested pathways, suggesting that offering more visible pathway options would be of significant interest to participating youth. For example, the most frequently earned badges required completion of 3-4 programs from the same category [3].

The goal of this poster is to introduce a scalable computational approach to provide pathways for youth, aiding their exploration of related or more advanced programs within the same category. CCOL hosts 20279 educational programs across 10 different categories, making it prohibitive to create pathways by hand. This initial work, in collaboration with the CCOL team at Northwestern University, will lead toward a mobile app with a built-in recommender system that uses program participation patterns, youth interests, and locations of users and programs to recommend programs for youth to deepen their interests. This poster will explore pathway creation in one category, 'Coding + Games', which has 698 programs, to show the promise of this approach.

2. RESEARCH METHODOLOGY

The process of pathway construction is composed of three components: extracting keywords, connecting programs based on their similarity into a network, and path discovery within the network. The set of keywords is determined by comparing the word frequencies of all program descriptions in a category to the word frequencies of the descriptions from all remaining categories. This comparison uses the Dirichlet prior model as used in[2], which is a probability distribution used to determine the relative importance of each keyword to both the text and the corpus of interest. We chose to use this model instead of more common LDA topic models because many descriptions in our corpus were too short to observe meaningful topic distributions. Fifty keywords are extracted from each category, which allows over 90% of the programs to be assigned at least one keyword. More keywords were not used, as the increase in program coverage per keyword decreases as the number of keywords increases.

To establish the connections between programs, the similarity between each pair of programs is calculated. Program similarity is computed as the cosine distance between two programs' keyword vectors. Each program is represented by a vector of length 50, one for each of the extracted keywords. For each element of the vector, if the program contains that keyword it is a 1, otherwise it will be 0. For programs to be related, they must share at least one keyword. Using these connections, a graph is then constructed to represent these relationships. The weight of an edge is the similarity between the nodes on either end, and each node is represented by the program identifier and the keywords of that program.

Tinker Studio
Hack Shop   Day 2
Youth Programs
Girls Do Hack 2014
Designed Objects: Toy Design/3D Model

Figure 1. Excerpt of a Learning Pathway

We use a graph search approach, based on the A* algorithm, to determine learning pathways through this network. Previous attempts to create pathways based on program difficulty proved ineffective due to the introductory nature of many of these programs, as well as the tendency for program providers to write descriptions for a broad audience. Our approach links programs based on their conceptual overlap, rather than difficulty levels. The challenge is to identify programs with some overlap, but not too much, so that youth are building on their previous experiences and interests. This encourages paths that cover all the topics in a category, resulting in numerous possible paths from any given starting point in the network. A user of this system would see the possible pathways available from a pathway once they have selected it to view or enroll.

3. RESULTS AND DISCUSSION

Our extraction approach yielded a set of keywords that were relevant to 92% of the programs in our sample, showing good coverage of the programs. Each of these programs contained 2.5 keywords on average, with the maximal program containing 11. While most of the keywords were relevant to the category, there were also ”noisy keywords”, which were either overly general (i.e., ”tech”) or irrelevant (i.e., ”board games”). This suggests that CCOL needs to provide better descriptions of the intent behind these high-level categories and that program providers could use additional support in crafting program descriptions to contain more accurate information about the nature of the program. To this end, we have worked with the CCOL team at Northwestern to develop a rubric for program providers to create descriptions that are more useful to both the youth using the system and the algorithm described here. At the time of writing there is limited new data to measure the effect of this tool.

Among the programs covered by the keywords, each was connected to an average of 25% of the programs in the category, with the maximally connected program being linked to 73% of the programs in the category. On the positive side, this is a sign of category coherence. On the negative side, this density of linkages can be attributed to the general nature of many of the program descriptions. Our graph search algorithm constructed multiple learning pathways for a given program in the network, with each path completing all keywords in the category. Figure 1 shows an excerpt from a constructed learning pathway. Most of these programs build towards a set of common ideas such as fashion, design, tinkering, and coding, resulting in a seemingly coherent pathway. However, there is an example that illustrate the problems with overly general descriptions: it is ambiguous what the Youth Programs offering is about, or how it fits into this pathway.

The length of the pathways generated is also a point of concern. Given the population using CCOL and the nature of the programs offered, pathways need to be short so as not to lose youth interest. Since the length of any path is dependent on the number of topics extracted from the category, a natural solution would be to limit the number of topics. This would decrease pathway length as well as eliminate some of the less useful keywords. However, the number of programs covered by the extracted topics drops off drastically as the number of topics decreases. This is concerning, as creating pathways to support interest deepening should not come at the cost of limiting the available program choices youth are offered. At time of writing, this problem is not addressed.

4. CONCLUSIONS

In this pilot study, we showed the promise of an automatic and scalable method for learning pathway creation. We successfully generated a representative set of keywords for a specific category of educational programs, and used these labels to create a similarity network between programs. We constructed multiple pathways through this network from a given starting point. The pathways created from this approach will inform a recommender system for youth using CCOL as well as provide youth with important information about how a new program may relate to their interests, helping support their interest-driven learning. The scalability of this approach is the key contribution of this work, as no fixed set of learning pathways will be appropriate for all types of learners, especially in connected and personalized learning environments such as CCOL. We hope to empirically demonstrate the effectiveness of this approach in the future through user studies.

Despite the promise of this method, it has some important limitations. Given the naive nature of the algorithm, the connections between programs and the pathways in the graph are only as meaningful as the keywords. In our work, we discovered multiple areas in which program description and the resulting keywords could be improved. Improving descriptions will help the performance of our algorithm, increasing users’ awareness of the programs they can choose.

5. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under grant number 1637350.

6. REFERENCES

[1] M. Ito, K. Gutiérrez, S. Livingstone, B. Penuel, J. Rhodes, K. Salen, J. Schor, J. Sefton-Green, and S. C. Watkins. *Connected learning: An agenda for research and design*. BookBaby, 2013.

[2] B. L. Monroe, M. P. Colaresi, and K. M. Quinn. Fightin’words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, 16(4):372–403, 2008.

[3] W. R. Penuel, T. Sumner, O. Dibie, M. A. Sultan, D. Quigley, N. Dadey, and K. Van Horne. Modeling the ecosystem of opportunity in the chicago cities of learning. 2015.

Non-Model based Global Item Discrimination Estimation using Deep Belief Network without Q-Matrix for Cognitive Diagnosis

Kang Xue
University of Georiga
kangxue@uga.edu

ABSTRACT

In this paper, we propose a methodology under the cognitive diagnostic measurement (CDM) to estimate global item discrimination without specifying the cognitive diagnostic model and the Q-Matrix. To achieve the estimation, we firstly design a deep belief network (DBN) based dimensional reduction framework to project high dimensional item response data to a low dimensional attribute space; secondly, by using the attribute estimates, the global item discrimination can be calculated. Unlike traditional model based method, only item responses and the total number of attributes are required for our methodology.

Keywords

CDM, deep belief network, item discrimination

1. INTRODUCTION

The purpose of cognitive diagnostic measurement (CDM) is to provide students' latent knowledge (attributes) mastery status through their responses to items from designed assessments, in other words, CDM can group students to different latent classes of which the attribute mastery status (attribute profiles) of group members are same. Due to the ability to provide educators the diagnostic feedback on students' assessment results, CDM has already attracted lots of research attention, and various types of diagnostic classification models (DCMs), such as DINA model, DINO model and LCDM, are designed based on different cognitive theories [2].

In classified test theory (CTT), item discrimination is a measure of the relationship between an item score and the total test score. An item is more discriminating if the examinees with high total score answer it correctly and the examinees with low total score answer it incorrectly. There is a similar concept can be used for DCMs. After obtaining the estimates of item parameters for a DCM, the goal of the item discrimination is to evaluate the diagnostic quality of items, which is how well does one item to differentiate between the examinees who master more attributes and the examinees who master fewer attributes. Let π_{i,α_h} denote the probabilities of a correct response to an item for examinees who have master more attributes, and π_{i,α_l} denote the probabilities of a correct response to an item for examinees who have master fewer attributes, so the generic discrimination index for item i in the context of LCDM can be defined as *p-value*: $d_i = \pi_{i,\alpha_h} - \pi_{i,\alpha_l}$. Item discrimination plays an very important role in diagnostic measurement because more items with high discrimination can achieve more accurate analysis results in contrast to items with lower discrimination [6].

There are two different definitions of item discrimination: global item discrimination and attribute-specific item discrimination. The global item discrimination is to compare the item performance of examinees master all required attributes and the examinees master none of the required attributes of one item. The attribute-specific item discrimination is to evaluate the item diagnostic quality for specific attribute. Global item discrimination is more alternative and attribute-specific item discrimination is more specific. In this paper, the item discrimination indicates global item discrimination.

To estimate global item discrimination, DCMs require both known specific models and correct Q-matrix, which indicates the relationship between items and attributes. However, in some cases, models are hard to determine before data analysis, and Q-matrices are misspecified or missing [5, 4]. To solve these two issues, we proposed an non-model based methodology to estimate global item discrimination using deep belief network (DBN) dimensional reduction method, which only requires item responses from students and the total number of attributes all items measure.

2. METHODOLOGY

Deep Belief Network (DBN) is a type of classic neural network and can be viewed as a composition of Restricted Boltzmann machine (RBM), which is generative stochastic artificial neural network that can learn a probability distribution over its set of inputs [1, 3]. In our framework, we assume that only the item responses and the number of attributes are known. Thus, in the DBN structure (Figure 1), the number of inputs equals to the number of item responses, the number of outputs equals to the number of attributes. The number of notes in hidden layer equals to (Number of inputs + outputs)/2. After constructing the DBN, the connection weights between two layers are estimated through minimize the following reconstruction errors using Stochas-

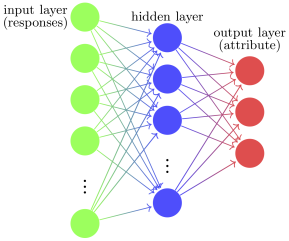


Figure 1: The structure of the DBN.

tic Gradient Descent (SGD):

$$
\begin{aligned}
&L(X, X') = ||X - X'||^2 \\
&X' = \sigma(W'_{hidden}(\sigma(W'_{output} Y + b'_{hidden}) + b_{input})) \qquad (1) \\
&Y = \sigma(W_{output}(\sigma(W_{hidden} X + b_{hidden}) + b_{output}))
\end{aligned}
$$

where $\sigma(.)$ is called sigmoid function, W' is the transpose of W, b and b' are the bias vectors, X indicates the inputs, X' is the reconstruction of inputs.

After training the DBN, when input a response vector from a student, the value of each output note indicates the probability this student mastery one attribute. Since the Q-Matrix is assumed unknown in our framework, this method cannot correspond one output note to a specific attribute. However, we can determine the following two latent classes according to the group mean of total scores (numbers of correct answers): the latent class $\mathcal{C}_0$ of which the members have the attribute profile vector $\vec{0}$, and the latent class $\mathcal{C}_1$ of which the members have the attribute profile vector $\vec{1}$. In other word, students in $\mathcal{C}_0$ master none of the required attributes, students in $\mathcal{C}_1$ master all of the required attributes. The average total score of $\mathcal{C}_0$ is the minimum among all latent classes, and the average total score of $\mathcal{C}_1$ is the maximum among all latent classes.

Let α_0 and α_1 indicate attribute profiles from $\mathcal{C}_0$ and $\mathcal{C}_1$ respectively. It is easy to know that for ith item, $\pi_{i,\alpha_0} \approx \pi_{i,\alpha_l}$, and $\pi_{i,\alpha_1} \approx \pi_{i,\alpha_h}$, where α_l is the attribute profiles that none of the attribute elements required by the ith item are mastered, and α_h is the attribute profiles that all of the attribute elements required by the ith item are mastered. Thus, the global item discrimination for ith item (*p-value*) can be calculated directly as following: $d_i \approx \pi_{i,\alpha_1} - \pi_{i,\alpha_0}$.

3. RESULTS

In this paper, we use simulated data to test the performance of our methodology. The data set is simulated under a general item by latent classes matrix [7], and it contains 5000 examinees and 40 items measure 3 attributes. First we compare the classification performance of the two latent classes $\mathcal{C}_0$ and $\mathcal{C}_1$ with two widely used DCMs, DINA and LCDM under two conditions: 1). assume the correct Q-Matrix is known for DCMs, 2). assume the Q-Matrix is missing for DINA and LCDM (all elements of Q-Matrix is randomized, the total error rate is 58%). All model based methods are conducted using "CDM package" in R. DBN method is accomplished using "tensorflow library" in Python. The F_1 score is used to quantify the quality of classification of the two latent classes $\mathcal{C}_0$ and $\mathcal{C}_1$. From the comparison results shown in Table 3, under the first condition, the $F_1(\mathcal{C}_0)$ using DBN is very close to the ones using DINA and LCDM, the $F_1(\mathcal{C}_1)$ using DBN is very close to LCDM but much higher than DINA. Since our DBN based method doesn't rely on Q-Matrix, the accuracy is same under the second condition. By contrast, both $F_1(\mathcal{C}_0)$ and $F_1(\mathcal{C}_1)$ for DINA and LCDM decrease significantly when Q-Matrix is misspecified.

Methods	F_1 $(\mathcal{C}_0)$	F_1 $(\mathcal{C}_1)$	F_1^* $(\mathcal{C}_0)$	F_1^* $(\mathcal{C}_0)$
DBN	.9514	.9637	.9514	.9637
DINA	.9545	.8684	.6363	.6591
LCDM	.9720	.9777	.7271	.7331

Table 1: The classification comparison. F_1 indicates the F_1 score under the first condition, F_1^* indicates the F_1 score under the second condition. Noting that $F_1 = F_1^*$ for DBN.

Secondly, we evaluate the global item discrimination (*p-value*) d_i by comparing with the traditional methods using item parameter estimates under DINA and LCDM. The criterion used to quantify estimation quality is the mean square error (MSE) of $\{d_i\}$ for all items. As shown in Table 2, the MSE of DBN method is much lower than the ones using DINA and LCDM with correct Q-Matrix.

Methods	**DBN**	DINA	LCDM
MSE	.0015	.0744	.0542

Table 2: Comparison of *p-value* estimation.

4. CONCLUSION

This paper proposes an non-model based method to evaluate global discrimination without using Q-Matrix through deep belief network. Our methodology first projects high dimensional responses vectors to low dimensional attribute space. Then global item discrimination can be calculated using the responses of two special latent classes ($\mathcal{C}_0$ and $\mathcal{C}_1$) which are estimated using DBN. The simulated experimental results show that our DBN method performs much better in classifying these two latent classes than using DINA and LCDM when Q-Matrix is unknown. Our methodology also provides more accurate global item discrimination than using item parameter estimates.

5. REFERENCES

[1] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT press, 2016.

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

[3] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[4] M. J. Madison and L. P. Bradshaw. The effects of q-matrix design on classification accuracy in the log-linear cognitive diagnosis model. *Educational and Psychological Measurement*, 75(3):491–511, 2014.

[5] A. A. Rupp and J. Templin. The effects of q-matrix misspecification on parameter estimates and classification accuracy in the dina model. *Educational and Psychological Measurement*, 68(1):78–96, 2007.

[6] A. A. Rupp, J. Templin, and R. A. Henson. *Diagnostic measurement: Theory, methods, and applications.* Guilford press, 2010.

[7] G. Xu and S. Zhang. Identifiability of diagnostic classification models. *psychometrika*, 81(3):625–649, 2016.

Non-Model based Attribute Profile Estimation with Partial Q-Matrix Information for Cognitive Diagnosis using Artificial Neural Network

Kang Xue
University of Georiga
kangxue@uga.edu

ABSTRACT

In this paper, we design a framework to estimate attribute without specific model for cognitive diagnosis measurement (CDM) using artificial neural network (ANN). In contrast to the previous research which relied on correctly specified Q-Matrix, our methodology only requires partial Q-Matrix information (simple items' q-vector). Simulated experiments are conducted to test the effect of three types of test factors to our method. The simulated study shows that our methodology provides an good option to analyze the data when the prior information is insufficient.

Keywords

CDM, ANN, Q-Matrix, attribute estimation

1. INTRODUCTION

The purpose of cognitive diagnostic measurement (CDM) is to group students to different latent classes based on a set of latent knowledges or attributes mastery status through their responses to items designed to measure these attributes. Students within the same latent class have same attribute profiles. In last two decades, CDM has attracted lots of research attention, and various of diagnostic classification models (DCMs), such as DINA, DINO and LCDM, are designed based on different types of cognitive theories [5].

Although different DCMs have various statistic structure, all recent DCMs are built with a fully probabilistic model structure rely on latent variables and and most research efforts have begun that represent a "step back" from the existing DCMs [9]. Another key concepts of current DCMs is Q-Matrix, which traditionally contains the items in the rows and the attributes in the columns, specifies which attributes are measured by each item. Each row of Q-Matrix is called q-vector. Simple item measures only one attribute, and complex item measures more than one attribute. Several researchers have already showed the effects of Q-Matrix misspecification to different types of diagnostic classification models from experiments [7, 8, 6].

With the rapid development of deep learning, in last several years, some research works started to introduce ANNs (MLP, SOM) to estimate students' attribute profiles to category students [2, 4]. However, there are still some limits of these researches: 1) correct Q-Matrix is assumed to be known; 2) the data is simulated under DINA model. In this paper, we propose an ANN based method to estimate students attribute profile only using partial Q-Matrix information. To accomplish our method, three test conditions are assumed to hold: 1) the number of attributes is known; 2) q-vectors of simple items are known; 3) at least one simple item with high discrimination[1] measure one attribute. All these three conditions are easy to be hold in real assessments.

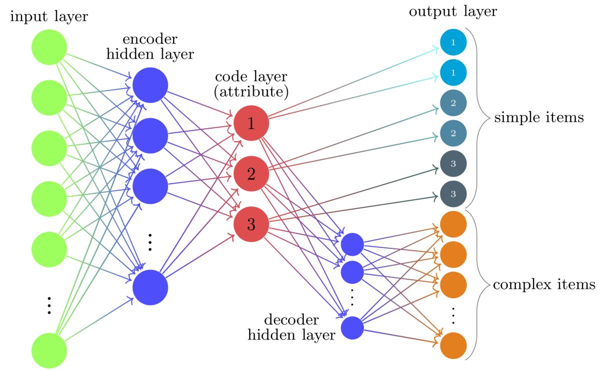


Figure 1: The structure of modified autoencoder.

2. METHODOLOGY

ANNs is a computational system inspired by biological neural systems for information processing in animals' brains. ANNs have showed competitive performance in large data set tasks because the universal approximation theorem behind neural networks [1]. In our proposed methodology, we designed a modified autoencoder network [3] to estimate students' attribute profiles using their item responses. The aim of an autoencoder is to learn a representation for a set of data, typically for the purpose of dimensionality reduction. As shown in Figure 1, our autoencoder contains two part:

[1] The goal of the item discrimination is to evaluate the diagnostic quality of items, which is how well does one item to differentiate between the students who master more attributes and the students who master fewer attributes.

encoder and decoder. Encoder contains input layer, encoder hidden layer and code layer, and decoder contains code layer, decoder hidden layer and output layer.

The number of inputs and outputs equal to the number of items I. The number of codes equals to the number of attributes A. The number of notes in encoder hidden layer equals to $H_e = (I + A)/2$. All the neurons in encoder are fully connected. Let $\mathcal{X}_i$ denote the ith neuron in input layer, $\mathcal{H}_h^e$ denote the hth neuron in encoder hidden layer and $\mathcal{C}_a$ denote the ath neuron in code layer. By given the input sets $\mathcal{X}$, the value of ath code neurons can be calculated through the following equation:

$$\begin{aligned}\mathcal{C}_a &= \sigma(\sum_{h=1}^{H_e} w_{h,a}\mathcal{H}_h^e + b_a) \\ &= \sigma(\sum_{h=1}^{H_e} w_{h,a}(\sigma(\sum_{i=1}^{I} w_{i,h}\mathcal{X}_i + b_h)) + b_a)\end{aligned} \tag{1}$$

where $w_{i,h}$ and $w_{h,a}$ are the connection weight from the ith input neuron to the hth hidden neuron and from the hth hidden neuron to the ath code neuron, b_h and b_a denote the bias for hidden neuron and code neuron respectively, and $\sigma(\cdot)$ indicates the sigmoid function.

After obtaining the values of code neurons, the decoder uses these values as input to reconstruct the item responses. Unlikely the typical decoder of which neurons are fully connected, we use a sparse connection strategy according to the partial know Q-matrix. Let $\mathcal{H}_h^d$, $\mathcal{X}_i'$ denote the hth neuron in decoder hidden layer and the ith neuron in output layer respectively. This strategy can be mathematically represented as following:

$$\mathcal{X}_i' = \begin{cases} \sigma(w_{a,i}\mathcal{C}_a + b_i), & i\text{th item is simple item} \\ \sigma(\sum_{h=1}^{H_d} w_{h,i}\mathcal{H}_h^d + b_i), & i\text{th item is complex item} \end{cases} \tag{2}$$

and $\mathcal{H}_h^d = \sigma(\sum_{a=1}^{A} w_{a,h}\mathcal{C}_a + b_h)$, where $w_{a,h}$ is the connection weight from the ath code neuron to the hth decoder hidden neuron, $w_{a,i}$ is the connection weight from ath code neuron to the ith output neuron, $w_{h,i}$ is the connection weight from the hth decoder hidden neuron to the ith output neuron, and b_h, b_i are biases of neurons. Noting that the number of neurons in output layer equals to the one in input layer, and the number of neurons in decoder hidden layers $H_d = (A + I_c)/2$, I_c is the number of complex items in the test.

After building up our modified autoencoder structure, the stochastic gradient descent (SGD) [3] is used to train this network using examinees' responses to items by minimizing the cost function $||\mathcal{X} - \mathcal{X}'||^2$. After training, once inputing eth examinee's response vector X_e into the autoencoder, a code vector $\hat{\alpha}_e$ can be obtained, each element of the $\hat{\alpha}_e$ indicates whether one attribute is mastered. For example, if $\hat{\alpha}_e = [1, 1, 0]$, the eth examinee masters attribute 1 and 2.

3. RESULTS

In this section, we design an experimental test using simulated data set to evaluate our framework under different assessment conditions which vary under 3 assessment factors: test length, number of simple items per attribute, test discrimination. Test length is the number of items contained in assessment. 10, 15 and 20 items are contained in short, medium and long test respectively. The number of simple items per attribute varies from 1 to 3. Because in real test, the proportion of simple items cannot be too high, so for short test, there are only two types of simple item proportions (1 or 2 simple items per attribute). Test discrimination indicates the proportion of items with high discrimination an assessment contains. Three levels, 50%, 70%, and 90% are set for low, medium and high test discrimination or test diagnostic quality. In contrast to some previous research, the response data is simulated using a item by latent class matrix [10], which is more general than the DINA model based simulation. The number of examinees is 2000, and the number of attributes in the test is 3.

First, we test the effects of the 3 assessment factors on attribute estimation. From Figure 2, we can find that 1) the marginal estimation accuracy for each attribute across simple items per attribute given test length is over 70%, which is under the condition that test length is 10 and only 1 simple item measures each attribute, and over 80% for half number of test conditions; when the test length of assessment and the number of simple items are sufficient, the accuracy is close to 90%; 2) if the test length is fixed, adding more simple items with Q-matrix to replace the complex item without Q-matrix can improve the estimation accuracy; 3) The last observation from Figure 2 is when fixing the number of simple items, adding more items without specified Q-matrix can still improve the estimation performance which is not available for latent classification model requiring Q-matrix.

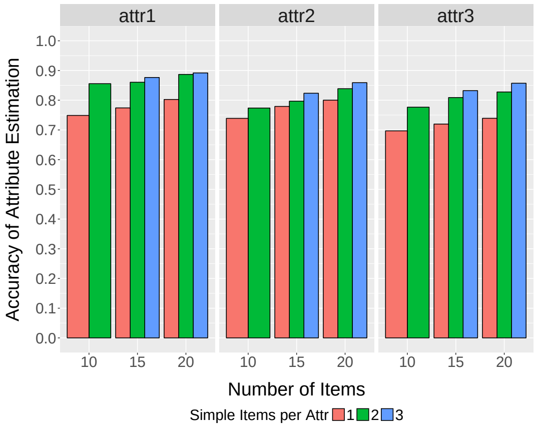


Figure 2: Marginal attribute estimation accuracy vs. number of simple items per attribute given test length.

From Figure 3, 1) we can get the similar overview like Figure 2, the marginal estimation accuracy for each attribute across the test discrimination is over 70%, and over 80% for half number of test conditions; when the test length of assessment and number of simple items are sufficient, the accuracy is close to 90%; 2) we can also observe that the test diagnostic quality has a positive correlation to the marginal estimation accuracy when fixing the test length; 3) another observation from Figure 3 is that our method has the power to improve the estimation accuracy just by adding items without knowing its q-vectors and discrimination levels, in other word, bad test items. For example, the short test with medium test discrimination, which contains total 7 items with high discrimination, has the same number of items with high discrimination contained by the medium test with low test discrimination. The medium test with low test discrimination has a more accurate estimation than the short test with medium test discrimination.

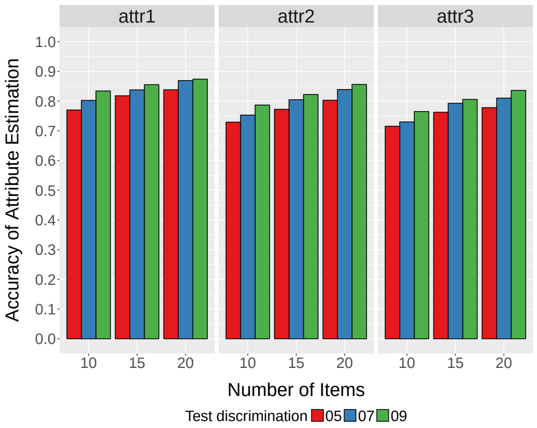


Figure 3: Marginal attribute estimation accuracy vs. test discrimination level given test length. *05*, *07* and *09* indicate low, medium and high test discrimination, respectively.

Secondly, we conduct a comparison between our method and the latent classification model based methods. Since when giving the correct Q-matrix and select appropriate model, latent model based methods always achieve the best estimation accuracy [2]. Thus in this part, we only compare the methods under the condition that Q-matrix is misspecified: the test which contains 20 items measure 3 attributes, and 12 of them are simple items. The diagnostic quality of this test is medium (70% of items are high discriminative), and we assume 50% elements of the Q-matrix for 8 complex items are misspecified, the Q-matrix for 12 simple items is correct, and the total misspecification rate is 20% for whole Q-matrix. Since the data are simulated under LCDM, we choose DINA (noncompensatory) and LCDM (compensatory, general) model for comparison. The model based estimation is accomplished using "*CDM package*" in R. The results are shown in Table 1. From Table 1, we can firstly find that LCDM achieves a better performance on both single attribute and attribute pattern estimation than DINA because it is a more general latent model. Secondly, we can observe that our proposed method shows the highest accuracy on attribute 1, attribute 3 and attribute pattern estimation; for attribute 2, the estimation accuracy is very close to LCDM model and higher than DINA model.

Models	attr 1	attr 2	attr 3	attr pattern
DINA	.901	.914	.900	.739
LCDM	.903	**.919**	.907	.756
ANN	**.909**	.916	**.931**	**.777**

Table 1: Comparison with model based method in estimation accuracy

4. CONCLUSION

The object of this paper is to propose a non-model based method with less constrains according to two potential issues in model based cognitive diagnosis: model selection and Q-matrix misspecification. To achieve this target, we design a modified autoencoder neural network for attribute estimation which doesn't rely on specific model assumption and just require partial Q-matrix information (simple items' q-vector). We test our methodology under different types of simulated test conditions according to test length, number of simple items per attribute, and the test diagnostic quality. The experimental results show that our methodology provides an option for users to analyze the data when lacking of prior knowledge about the items. Another advantage of this methodology showed in experimental results is that even adding "bad" items without correct q-vector, this method can improve the estimation accuracy.

5. REFERENCES

[1] B. C. Csáji. Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University*, 2001.

[2] Y. Cui, M. Gierl, and Q. Guo. Statistical classification for cognitive diagnostic assessment: An artificial neural network approach. *Educational Psychology*, 36(6):1065–1082, 2016.

[3] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[4] Q. Guo, M. Cutumisu, and Y. Cui. A neural network approach to estimate student skill mastery in cognitive diagnostic assessments. 2017.

[5] R. Henson, J. Templin, and J. Willse. Defining a family of cognitive diagnosis models using log linear models with latent variables. *Psychometrika*, 74:191–210, 2009.

[6] O. Kunina-Habenicht, A. A. Rupp, and O. Wilhelm. The impact of model misspecification on parameter estimation and item-fit assessment in log-linear diagnostic classification models. *Journal of Educational Measurement*, 49(1):1–23, 2012.

[7] M. J. Madison and L. P. Bradshaw. The effects of q-matrix design on classification accuracy in the log-linear cognitive diagnosis model. *Educational and Psychological Measurement*, 75(3):491–511, 2014.

[8] A. A. Rupp and J. Templin. The effects of q-matrix misspecification on parameter estimates and classification accuracy in the dina model. *Educational and Psychological Measurement*, 68(1):78–96, 2007.

[9] A. A. Rupp, J. Templin, and R. A. Henson. *Diagnostic measurement: Theory, methods, and applications.* Guilford press, 2010.

[10] G. Xu and S. Zhang. Identifiability of diagnostic classification models. *psychometrika*, 81(3):625–649, 2016.

Bayesian Partial Pooling to Improve Inference Across A/B Tests in EDM

Adam C Sales
University of Texas at Austin
536C George I. Sánchez Building
Austin, TX 78705
asales@utexas.edu

Thanaporn Patikorn
Worcester Polytechnic Institute 100 Institute Rd
Worcester, MA 01609
tpatikorn@wpi.edu

Neil T. Heffernan
Worcester Polytechnic Institute 100 Institute Rd
Worcester, MA 01609
nth@wpi.edu

ABSTRACT

This paper will explain how analyzing experiments as a group can improve estimation and inference of causal effects–even when the experiments are testing unrelated treatments. The method, composed of ideas from meta-analysis, shrinkage estimators, and Bayesian hierarchical modeling, is particularly relevant in studies of educational technology. Analyzing experiments as a group–"partially pooling" their respective datasets–increases overall accuracy and avoids issues of multiple comparisons, while incurring small bias. The paper will explain how the method works, demonstrate it on a set of randomized experiments run within the ASSISTments platform, and illustrate its properties in a simulation study.

1. INTRODUCTION

Using educational technology to conduct many experiments, as in the ASSISTments TestBed [7], allows education researchers to rigorously answer many causal questions and test many hypotheses independently. Perhaps more surprisingly, the various experiments can help *each other*. Effect estimates that partially pool data across experiments—even those that are testing very different interventions—are often more precise and accurate, and less error-prone, than estimates based on the experiments individually.

This poster will illustrate a Bayesian approach to analyzing several experiments simultaneously. (By "Bayesian," here, we mean merely that the goal of the approach is a posterior distribution for treatment effects.) The method combines ideas from [8] and [1] on shrinkage, from [12] on Bayesian partial pooling to examine treatment effect heterogeneity, and [6] on multiple comparisons. The paper's main contributions will be to introduce these ideas to an EDM audience—where, due to proliferation of online experiments, they are particularly applicable—and to illustrate their potential.

Previously, [10] combined data across experiments to improve covariance adjustment; that method is orthogonal, and perhaps complementary, to ours, which does not use covariates. [3], [9], and many others have used multilevel, hierarchical Bayesian modeling to analyze intelligent tutor data, but not in the context of experiments.

After describing and explaining the method (Section 2), we will illustrate it in an analysis of a dataset comprised of 22 parallel experiments run inside ASSISTments [13] (Section 3) and in a simulation study (Section 4). We will show that partially pooling data from across experiments increases precision while lowering type-I error rates, decreases the width of confidence intervals while improving their coverage, and substantially reduces the incidence of drawing incorrect conclusions from experimental data.

2. SHRINKAGE, PARTIAL POOLING, AND REGRESSION TO THE MEAN

Unbiased estimates $\hat{d}^{np}$ of effect sizes d from randomized A/B tests are noisy—a different estimate would have resulted had the treatment been randomized differently. The standard error of a particular effect size estimate, $\sigma_i = SD(\hat{d}_i^{np}|d_i)$, depends on a number of factors, most principally the sample size n_i, but in practice it is never zero. Similarly, among a group of K experiments, the true effect sizes d_i, $i = 1, ..., K$, (presumably) vary as well—$var(d) = \tau$, say. Considered together, the variance of a group of effect size estimates is the sum of both components: the variance of the true effects plus the average of the (squared) standard errors of the individual estimates:

$$var(\hat{d}^{np}) = \tau^2 + \mathbb{E}[\boldsymbol{\sigma}^2]$$

In other words, the distribution of effect size *estimates* is wider than the distribution of true effect sizes. Therefore, the largest effect size estimates $\hat{d}^{np}$ typically *over*estimate their respective true effects d, and that the smallest effect size estimates typically *under*estimate their true effects. This is an example of regression to the mean [4] (also see [16]).

The implication for estimating effects can be startling. When A/B tests are analyzed independently, the best estimate for the true effect size d_i in experiment i is $\hat{d}_i^{np}$. However, when the K experiments are considered as a group, $\hat{d}^{np}$ is inadmissible. A better estimate, $\hat{d}^{pp}$, corrects for the fact that the extreme estimates are probably too extreme, and shrinks them toward the overall mean effect size μ [2]:

$$\hat{d}_i^{pp} = \mu + c_i \left(\hat{d}_i^{np} - \mu \right) \quad (1)$$

where c_i is a "shrinkage coefficient" between 0 and 1. When $c_i = 1$, $\hat{d}_i^{pp} = \hat{d}_i^{np}$; when $c_i = 0$, $\hat{d}_i^{pp} = \mu$.

Another term for this procedure is "partial pooling" [5]. The overall mean treatment effect, μ, can be estimated by completely pooling the data across all K experiments. In contrast, individualized estimates $\hat{d}^{np}$ result if data from different A/B tests are not pooled at all—$\hat{d}^{np}$ is a "no pooling" estimate. The optimal estimate $\hat{d}^{pp}$ combines the the no-pooling estimate $\hat{d}^{np}$ with a complete-pooling estimate of μ—hence, partial pooling.

In general, the size of the shrinkage coefficient c_i, which regulates the extent of the partial pooling, depends both on the standard deviation of the true effects, τ, and σ_i, the standard error of $\hat{d}_i^{np}$. When τ is large, the experiments differ widely from each other, so the overall mean effect μ tells us little about the individual effects d. When σ_i is large, then $\hat{d}_i^{np}$ is quite noisy, and tells us little about d_i. The shrinkage coefficient c_i balances these two factors.

For instance, Rubin [12] models each $\hat{d}_i^{np}$ as normal, with mean d_i (since it is unbiased) and standard error σ_i:

$$\hat{d}_i^{np} \sim \mathcal{N}(d_i, \sigma_i) \quad (2)$$

This would be approximately the case if estimators $\hat{d}^{np}$ were difference-in-means or regression estimators from sufficiently large experiments. Then, he models the effects themselves as drawn from a normal distribution:

$$d_i \sim \mathcal{N}(\mu, \tau). \quad (3)$$

Under model (2)–(3),

$$c_i = \frac{\tau^2}{\tau^2 + \sigma_i^2}. \quad (4)$$

When τ is large (so the true effects are very different from each other) and σ_i is small (so $\hat{d}_i^{np}$ is very precise), c_i is close to one—the partial pooling estimator $\hat{d}_i^{pp} \approx \hat{d}_i^{np}$—data are barely pooled across experiments at all. Conversely, when σ_i is large (so $\hat{d}_i^{np}$ is noisy) and τ is small (so the true effects are similar to each other), then c_i is close to zero, and $\hat{d}^{pp} \approx \mu$, the overall mean effect size, completely pooling data across experiments. In general c_i is in between zero and one, and the estimator $\hat{d}^{pp}$ partially pools information between the individual effect estimate $\hat{d}_i^{np}$ and the overall mean μ. The mean of the true effects μ and their variance τ are, of course, unknown, but they may be estimated from the data.

Unlike $\hat{d}_i^{np}$, $\hat{d}_i^{pp}$ is biased—it is shrunk towards the overall mean μ. To compensate for the bias, $\hat{d}_i^{pp}$ is less noisy than $\hat{d}_i^{np}$; its standard error is $\sqrt{c_i}\sigma_i$. Since $c_i < 1$, this is always less than $\hat{d}_i^{np}$'s standard error σ_i. Overall, [15] shows the root mean squared error (RMSE) of the estimates $\hat{d}^{pp}$, considered as a group, will be less than the RMSE of the individual unbiased estimates $\hat{d}^{np}$. This result is that it applies even when the causal estimates do not need to be related in any way.

When analyzing a set of A/B tests run inside intelligent tutors, estimates of the effects based on partial pooling will be more accurate, on average, than estimates that consider each test individually.

3. ANALYZING 22 EXPERIMENTS

How does partial pooling work in practice, in an authentic EDM setting?

The ASSISTments TestBed [7] allows education researchers to propose and conduct minimally-invasive A/B tests within the ASSISTments intelligent tutor. The TestBed infrastructure automatically publishes anonymized data from these experiments. Conveniently, [13] combined 22 of these datasets into one publicly available file. All 22 experiments were skill builders, which are problem sets designed to teach, or bolster, a specific topic or skill. Inside a skill builder, students are required to solve problems associated to that skill until mastery is achieved, typically defined as answering three questions in a row.

The dataset includes a number of student features and two dependent measures. In this paper, We will focus only on one dependent measure `complete`, a binary variable indicating completion of the skill builder, taking value 1 if the student achieved mastery or 0 if the student either stopped working before achieving mastery or exhausted all of the skill builder's problems without achieving mastery.

To estimate treatment effects conventionally, without pooling across experiments, we fit a separate logistic regression to each of the 22 experiments, regressing `complete` on an indicator for treatment condition.

$$Pr(complete = 1) = invLogit\left(\alpha_{expr} + \beta_{expr} Z\right) \quad (5)$$

Where $invLogit(\cdot)$ is the inverse logit function. The intercept α_{expr} and treatment effect β_{expr} (the log odds ratio of completion for the treatment vs the control condition) were estimated separately in each experiment $expr$.

To estimate effects using partial pooling, we re-fit (5) within a Bayesian multilevel logistic regression using the `rstanarm` package [14] in `R` [11]. That is, we assigned models $\alpha_{expr} \sim \mathcal{N}(\alpha_0, \sigma_\alpha)$ and $\beta_{expr} \sim \mathcal{N}(\beta_0, \tau)$, where hyperparamters α_0, β_0, σ_α and σ_β were estimated from the data using weakly-informative priors.

Figure 1 plots estimated treatment effects and approximate 95% confidence intervals ($\pm 2SE$) for the 22 experiments, using both the conventional no-pooling estimator and the partially-pooling estimator. The partial pooling shrunk the estimates quite a bit: while the no-pooling estimates ranged from approximately -1.3 to 0.6, the partial pooling estimates were all close to zero, ranging from -0.2 to 0.1. The estimated standard errors were also much smaller for the partially pooled estimators. The average standard error for the no-pooling estimates was 0.39, whereas the average standard error for the partial-pooling estimates was less than half that, 0.17. Finally, though two of the no-pooling estimates were statistically significant, with confidence intervals excluding zero, none of the partial-pooling estimates was.

Figure 2 plots the estimated standard errors from the two

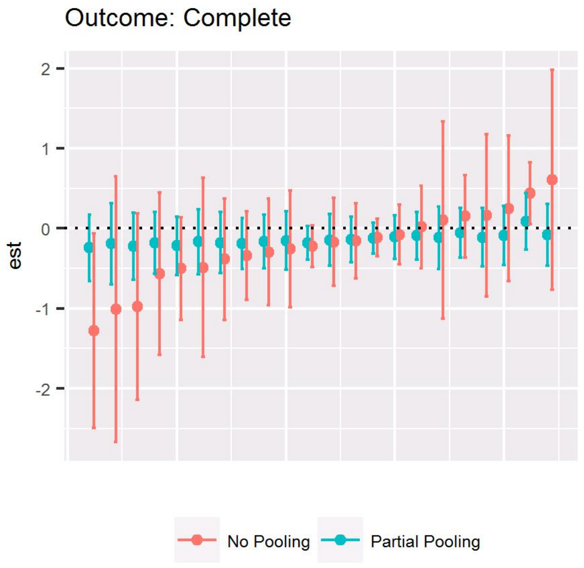


Figure 1: Partial-pooling an no-pooling treatment estimates and approximate 95% confidence intervals for the 22 experiments, arranged horizontally by the no-pooling treatment effect. The outcome was `complete`, and the treatment effects are log-odds ratios.

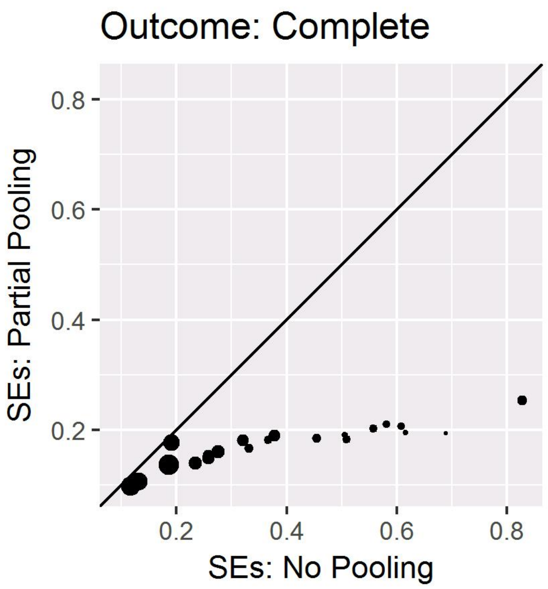


Figure 2: Partial-pooling vs no-pooling standard errors, with point size proportional to sample size in the experiment.

sets of estimates. The sizes of the points in the plot are proportional to experimental sample sizes. The partial-pooling standard errors are all smaller than those from the no-pooling estimates. However, the differences are not uniform. Experiments with large sample sizes and low no-pooling standard errors had partial-pooling standard errors that were only slightly smaller. As the sample sizes shrunk, both sets of standard errors grew. However, the no-pooling standard errors grew much faster. The largest difference in standard errors between the two methods was for studies with the smallest samples and the largest no-pooling standard errors.

4. A SIMULATION STUDY

Partial pooling worked as advertised when applied to the ASSISTments dataset, shrinking estimates towards zero and reducing standard errors, sometimes drastically—but did it get the right answers?

We ran a simulation study to investigate the performance of the partial-pooling estimator when the right answer is known.

4.1 Data Generating and Analysis Models

We simulated batches of $K = 20$ experiments each. Within a batch, sample sizes n varied from 20 to 115. Treatment Z was randomized to half of the subjects in each experiment. For each batch, outcomes Y were generated as

$$Y_i \sim \mathcal{N}(\alpha_{expr[i]} + \beta_{expr[i]} Z_i, \sigma_Y) \qquad (6)$$

with random intercepts $\alpha_{expr} \sim \mathcal{N}(0, 1)$ and treatment effects $\beta_{expr} \sim \mathcal{N}(0, \tau)$, both varying at the experiment level. The between-experiment standard deviation of treatment effects τ varied between runs. It took the values of $\tau = 0$, corresponding to $\beta_{expr} \equiv 0$ across all experiments, and $\tau = \{0.1, 0.2, 0.5, 1.0\}$. When τ was positive but low, there was a treatment effect in every experiment, but nearly all effects were very small. Larger values of τ corresponded to more variance in the treatment effects, including some that were substantial. For every study, $\sigma_Y = 1$.

The 20 experiments in each batch were analyzed both separately, with no-pooling estimators, and jointly, with a partial-pooling estimator. Both estimators fit model 6 to each dataset to estimate treatment effects β_{expr}; however, the partial-pooling estimator additionally modeled $\beta_{expr} \sim \mathcal{N}(\beta_0, \tau)$ and $\alpha_{expr} \sim \mathcal{N}(\alpha_0, \sigma_\alpha)$.

For each value of τ we ran 500 iterations of 20 experiments each, producing 10,000 experimental datasets.

4.2 Simulation Results

Table 1 gives the results of the the simulation. The estimated standard errors and root mean squared errors of partial pooling estimates were consistently substantially lower than those of no-pooling estimates—partial pooling increased both accuracy and precision. The differences between the estimators diminished as the variance of true treatment effects, τ increased. This is predicted by (4): as τ increases relative to no-pooling standard errors σ, the shrinkage coefficient tends towards 1 and the the partial pooling estimate tends towards the no-pooling estimate. Intuitively, when τ increases various experiments become less informative about each other, so partial pooling decreases in value.

		τ				
	Pooling	0	0.1	0.2	0.5	1
SE	Partial	0.13	0.14	0.17	0.23	0.25
	None	0.26	0.27	0.26	0.27	0.26
|Bias|	Partial	0.00	-0.05	-0.08	-0.08	-0.05
	None	0.00	-0.00	0.00	0.00	0.00
RMSE	Partial	0.09	0.12	0.17	0.24	0.26
	None	0.28	0.27	0.27	0.28	0.27
Coverage	Partial	1.00	0.98	0.95	0.95	0.95
	None	0.95	0.95	0.95	0.95	0.95

Table 1: Average standard error (SE), bias magnitude, root mean squared error (RMSE), and empirical coverage of 95% confidence intervals (Coverage) for partial pooling and no pooling estimates for different values of τ.

Table 1 also shows that while the no-pooling estimates are unbiased, the partial pooling estimates are slightly biased towards zero, as expected, with the bias decreasing as τ increases. This bias does not cause undercoverage of 95% confidence intervals. Remarkably, for low τ, the partial pooling confidence intervals *over*-covered—more than 95% of the realized confidence intervals included the true parameter. The width of the confidence interval is four times the standard error, by construction—so partial-pooling confidence intervals were both substantially smaller and more often correct.

5. DISCUSSION

Partial pooling is a surprising, and surprisingly effective, technique to improve education sciences in the big data era. As educational technology allows A/B testing to proliferate, partial pooling is a method to use some of the oldest results in statistics—such as regression to the mean—alongside new Bayesian technology to improve the precision and accuracy of experimental estimates. When experiments can be analyzed in a group, the result is smaller confidence intervals with the same or higher coverage.

Partial pooling is a model based technique, and it remains to be seen how it performs when the model is severely misspecified. A host of Bayesian model checking procedures, including some suggested in [12], may be brought to bear on this question. In any event, most effect estimates are approximately normally distributed, by the central limit theorem, so methods based on normal theory will apply.

All code and data for this paper may be found at `https://github.com/adamSales/EDMpartialPooling`*.*

6. REFERENCES

[1] B. Efron and C. Morris. Stein's estimation rule and its competitorsâĂŤan empirical bayes approach. *Journal of the American Statistical Association*, 68(341):117–130, 1973.

[2] B. Efron and C. Morris. Stein's paradox in statistics. *Scientific American*, 236(5):119–127, 1977.

[3] M. Feng, N. T. Heffernan, and K. R. Koedinger. Predicting state test scores better with intelligent tutoring systems: developing metrics to measure assistance required. In *International conference on intelligent tutoring systems*, pages 31–40. Springer, 2006.

[4] F. Galton. Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 15:246–263, 1886.

[5] A. Gelman and J. Hill. *Data analysis using regression and multilevel/hierarchical models.* Cambridge university press, 2006.

[6] A. Gelman, J. Hill, and M. Yajima. Why we (usually) don't have to worry about multiple comparisons. *Journal of Research on Educational Effectiveness*, 5(2):189–211, 2012.

[7] N. T. Heffernan and C. L. 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.

[8] W. James and C. Stein. Estimation with quadratic loss. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 361–379, 1961.

[9] Z. A. Pardos, M. Feng, N. T. Heffernan, and C. Linquist-Heffernan. Analyzing fine-grained skill models using bayesian and mixed effects methods. *Educational Data Mining*, page 50, 2007.

[10] T. Patikorn, D. Selent, N. T. Heffernan, J. E. Beck, and J. Zou. Using a single model trained across multiple experiments to improve the detection of treatment effects. In *Proceedings of the 10th International Conference of Educational Data Mining*, 2017.

[11] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2016.

[12] D. B. Rubin. Estimation in parallel randomized experiments. *Journal of Educational Statistics*, 6(4):377–401, 1981.

[13] D. Selent, T. Patikorn, and N. Heffernan. Assistments dataset from multiple randomized controlled experiments. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pages 181–184. ACM, 2016.

[14] Stan Development Team. rstanarm: Bayesian applied regression modeling via Stan., 2016. R package version 2.13.1.

[15] C. Stein. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 197–206, Berkeley, Calif., 1956. University of California Press.

[16] S. M. Stigler. The 1988 neyman memorial lecture: a galtonian perspective on shrinkage estimators. *Statistical Science*, pages 147–155, 1990.

Starters and Finishers: Predicting Next Assignment Completion from Student Behavior During Math Problem Solving

Taylyn Hulse
Worcester Polytechnic Institute
trhulse@wpi.edu

Avery Harrison
Worcester Polytechnic Institute
aeharrison@wpi.edu

Korinn Ostrow
Worcester Polytechnic Institute
ksostrow@wpi.edu

Anthony Botelho
Worcester Polytechnic Institute
abotelho@wpi.edu

Neil Heffernan
Worcester Polytechnic Institute
nth@wpi.edu

ABSTRACT

A substantial amount of research has been conducted by the educational data mining community to track and model learning. Previous work in modeling student knowledge has focused on predicting student performance at the problem level. While informative, problem-to-problem predictions leave little time for interventions within the system and relatively no time for human interventions. As such, modeling student performance at higher levels, such as by assignment, may provide a better opportunity to develop and apply learning interventions preemptively to remedy gaps in student knowledge. We aim to identify assignment-level features that predict whether or a not a student will finish their next homework assignment once started. We employ logistic regression models to test which features best predict whether a student will be a “starter” or a “finisher” on the next assignment.

Keywords

Student Modeling, Mathematics Education, Classification

1. INTRODUCTION

Online learning environments, paired with educational data mining research, provide student and teacher supports for learning. These environments are able to map student learning and behavior to personalize content, offer scaffolding, and provide real time support such as informational or motivational messages [11], making them nearly as effective as one-on-one human tutoring [14]. While great strides have been made in refining online learning environments to optimize learning, past work has primarily focused on student learning at the problem-level or using problem-level features within learning systems [3, 4]. These models provide immediate feedback to students and personalize learning within a user’s session. Less work has been done at higher granularities, such as modeling learning from assignment to assignment, to capture broader models of student learning.

While problem-level models of student learning are important, teachers more often care about higher level aspects of student learning such as whether students will be able to complete their homework assignment and if not, why? Building on previous work to track learning in online learning environments, as well as studies that have utilized similar data, we present our first attempt to build interpretable, predictive models of next-assignment completion. These models should indicate the best predictors of next-assignment completion to interpret reasons that a student might be a “starter” who is unable to finish the next homework assignment rather than a “finisher” who will complete the next assignment.

2. LITERATURE REVIEW

Online learning environments and tutoring systems contain rich data that can be applied to any level of fine- or coarse-grained research questions pertaining to student learning and behavior [9]. Using data from online systems, researchers have modeled student learning at various levels to better understand predictive behaviors, affective states, and system features of learning. From skill-level within problems [10], to problem-level [5], and across topics [2], the educational data mining community has tracked student learning and performance in a variety of contexts. Though steady progress has been made in predicting low-level behaviors, we can also leverage the prediction power of student logs to predict higher level behaviors and outcomes [1].

Research has also turned to predicting negative student behaviors and outcomes, such as student dropout rate. For instance, modeling student dropout rates has been a focus within massive open online courses (MOOCs) to understand why students complete online courses or dropout along the way [13, 16]. Similarly, attritional behavior in MOOCs has been modeled to identify and intervene with students who appear to be most likely to “stopout” [7]. While tutoring systems developed for K-12 curricula differ from MOOCs and secondary education settings, modeling dropout rates in online assignments would be beneficial at the K-12 level. Drawing from this work, we intend to develop predictive models of assignment dropout in an online learning environment to identify students likely to dropout of future assignments with time to intervene.

To accomplish this, we will use ASSISTments, a free, web-based tutoring system for K-college curricula that primarily features middle school mathematics content [8]. The current project will focus on “Skill Builders”, which are pre-built problem sets that map onto content areas to provide students with practice on topics featured on standardized tests. Skill Builders present problems from a given content area in a randomized order and are designed to challenge a student until that student achieves content mastery.

Table 1. Descriptive statistics of predictive features of assignment completion

Predictors	*N*	Minimum	Maximum	Mean	Standard Deviation
Current Assignment Completed	71,523	0	1	0.93	0.26
Assignment Mastery: 3-4 Problems	52,896	0	1	0.69	0.46
Assignment Mastery: 5-8 Problems	16,471	0	1	0.21	0.41
Assignment Mastery: 9+ Problems	2,156	0	1	0.03	0.17
Assignment Dropout: 0 Problems	3,750	0	1	0.05	0.22
Assignment Dropout: <4 Problems	1,122	0	1	0.01	0.12
Assignment Dropout: 4+ Problems	789	0	1	0.01	0.10
Average Attempt Count per Problem	77,084	0	2.91	1.15	0.42
Average Hint Count per Problem	75,130	0	1	0.59	0.18

Under default settings, students must consecutively answer three problems correctly to achieve mastery status for the assignment.

Previous research using ASSISTments and Skill Builders has sought to detect and fine-tune features of ASSISTments to be most beneficial to students and educators in practice. Most notably, a recent efficacy trial found that students who used ASSISTments throughout the school year performed better on an end-of-year standardized test than their counterparts who continued to use pen-and-paper homework assignments [12]. The researchers theorized that the difference in achievement may have been attributed to teacher reports generated in ASSISTments that provide teachers with information and timely homework feedback for students. This suggests that providing predictive feedback to teachers may better prepare them to provide additional support needed before difficult assignments. Predictive reports on future assignments would enable teachers to target specific content and assist students beforehand.

3. CURRENT PROJECT

We build off previous work that has modeled next problem performance at the student level, as well as approaches to modeling dropout rates in secondary education settings, to build predictive models of completion at the assignment level. While problem-level predictive models are limited to immediate, online-tutor interventions, making predictions about student behavior at the assignment level would allocate time for teacher interventions.

We use logistic regression to predict whether a student will be a "starter" or a "finisher" on the next ASSISTments assignment. Specifically, if a student opens the next assignment, will he be able to complete that assignment, achieving content mastery? We use the predictive model to identify features most predictive of next assignment completion. We focus first on using current completion features to predict next assignment completion and then progress to the predictive abilities of student behavior metrics within the tutor. This work is guided by the following three research questions: *How well does student completion on the current assignment predict completion on the next assignment? Does the number of completed problems matter when predicting next assignment completion? Do student behaviors within the tutor predict next assignment completion above current assignment completion features?*

4. DATA AND PREPROCESSING

We used publically available ASSISTments data [6] from Skill Builders during the 2016-2017 school year. Skill Builders were restricted to mathematics content and mastery parameters of 3+ correctly solved, consecutive problems. Using descriptive statistics, outliers were trimmed from any variables with a skewness statistic > |3|. After cleaning, the dataset contained 77,200 cases of assignments started or completed by 9,231 students in grades 3-12 across 5,143 unique Skill Builder assignments. Students completed the current assignment 92.65% of the time and completed the next assignment, our outcome variable, in 88.08% of cases. We constructed nine features from the dataset using assignment-level and problem-level variables aggregated by student at the assignment level (Table 1). In addition to current assignment completion, we selected assignment mastery speed, average attempt count per problem, and average hint count per problem as features. Assignment mastery speed is based on the number of problems students solve to fulfill the requirements of the skill builder assignment and "master" the content.

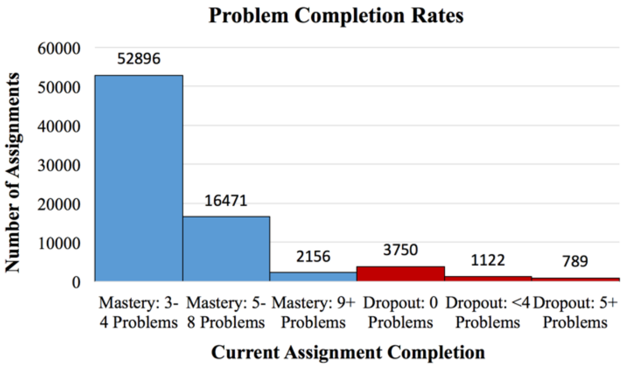


Figure 1. Rate of problem completion within complete and incomplete assignments.

We created three categories of mastery speed based on a method previously used [17]. We extended the method to account for the duration of student persistence prior to dropping out of the assignment, which we use as a dropout rate (Figure 1). To find a predictive model of future homework completion that optimized simplicity and fit, we created three models through logistic regression to compare and evaluate. We began with the completion model, with only current assignment completion as a predictor of next assignment completion. Then, we delved into student behavior within current assignments to build the binned completion model with completion status and number of problems completed within the assignment as predictors. Lastly, the binned completion and student features model used the binned completion predictors, the average hint use per problem, and average attempt count per problem as features. We used optimized thresholds for kappa and accuracy since our models were biased towards the majority class. By optimizing the threshold, we removed this bias. The optimized threshold was calculated only on training data. Five-fold cross validation was applied at the student level for each model.

5. RESULTS

Table 2 overviews the performance for each of the three models. Each model performs above chance (AUC > 0.50; kappa > 0.00) though performance slightly decreases (kappa, F_1, Accuracy) and AUC slightly increases as the models increase in complexity.

Table 2. Model comparison across performance metrics.

Model	AUC	Kappa	F_1	Accuracy
Completion	0.617	0.281	93.04	87.42
Mastery & Dropout	0.632	0.258	91.82	85.44
Mastery, Dropout, & Student Features	0.641	0.250	91.41	84.79

Model 1: Completion Model

The first model was a logistic regression that used completion on the current assignment to predict completion on the next assignment. This served as a base rate model that simply consists of completion as the predictor for future completion. For the Completion Model (kappa=0.281, AUC=0.617), completion was a positive predictor of next assignment completion, $\boldsymbol{b}$= 2.10, $z(77{,}200) = 71.05$, $p < 0.01$.

Model 2: Mastery and Dropout Model

The second model expanded on the base rate model by categorizing completeness and number of problems solved. We applied logistic regression using completeness and incompleteness categories as features to predict next assignment completion. Mastery speeds were significant, positive predictors of next assignment completion while dropout rates were not statistically significant (Table 3).

Table 3. Logistic regression with Mastery Speeds and Dropout Rates.

Feature	b	B	SE	z value	p value
Intercept	0.51	2.13	0.00	175.35	0.00
Mastery (3-4)	1.83	0.85	0.52	3.53	0.00
Mastery (5-8)	1.69	0.69	0.52	3.27	0.00
Mastery (9+)	1.26	0.21	0.52	2.43	0.02
Dropout (0)	-0.46	-0.10	0.52	-0.88	0.38
Dropout (1-3)	-0.10	-0.01	0.52	-0.20	0.84
Dropout (4+)	-0.05	-0.01	0.52	-0.10	0.92

Model 3: Mastery, Dropout and Student Features Model

The final model incorporates two student-related features: hints and attempts. We applied logistic regression using completeness and incompleteness categories, as well as the two student features, to predict next assignment completion. Table 4 shows that in addition to mastery speeds, average attempts was also a significant, negative predictor of next assignment completion. This suggests that more attempts in the current assignment results in a lower likelihood of finishing the next assignment.

Table 4. Logistic regression with Mastery Speed and student features.

Feature	b	B	SE	z value	p value
Intercept	0.67	2.13	0.00	3.53	0.00
Mastery (3-4)	1.82	0.84	0.52	3.52	0.00
Mastery (5-8)	1.75	0.72	0.52	3.38	0.00
Ave Attempt Count	-1.40	-0.06	0.04	-3.92	0.00
Mastery (9+)	1.34	0.22	0.52	2.57	0.01
Dropout (0)	-0.58	-0.13	0.52	-1.12	0.26
Ave Hint Count	-0.02	-0.00	0.07	-0.29	0.77
Dropout (4+)	0.07	0.00	0.52	0.14	0.89
Dropout (1-3)	-0.06	-0.00	0.52	-0.11	0.91

6. DISCUSSION

The models presented in this paper predict next assignment completion, which compared to predicting next problem completion, could provide more timely and practical information about student learning that could be applied through teacher intervention. Though most of the performance measures slightly decreased as more features were added, all three models performed similarly as a whole. Out of the student features we analyzed, completeness on the current assignment is the most prominent predictor of completion on the next assignment, which answered our first research question. This suggests that a simple model using only completeness as a predictor would be appropriate for uses such as creating an alert in a teacher dashboard to signal when students may not complete their next assignment.

That said, Models 2 and 3 add a more detailed explanation regarding how completeness breaks down and what other features may contribute to next assignment completion. We answered our second research question with Model 2 by categorizing completeness into mastery speed and dropout rate based on how many problems students completed. Though dropout rates were not significant predictors, higher mastery speeds in the current assignment increased the likelihood of students completing their next assignment. This is to be expected, as students who complete their current assignment in fewer problems are generally performing more efficiently, which may be suggestive of future performance due to underlying knowledge levels, motivational and behavioral tendencies, or other student-level characteristics.

To answer our third research question, Model 3 incorporated within-problem student behaviors, average attempts made, and average hints used per problem. The number of attempts was a negative predictor of next assignment completion, suggesting that students who make more attempts per problem are less likely to complete the next assignment. It seems that lower performing students (based on those who finish the assignment in more problems and take more attempts per problem) are less likely to complete the next assignment. While this is not a surprising finding, it brings us closer to teasing apart the integral facets of students' knowledge, behavior, and interaction with the system which leads them to become starters or finishers on their next assignment.

Though the models presented herein serve as a valuable first step towards understanding student and system contributions to student learning at a higher level, we acknowledge limitations in our dataset and analyses. We had a limited sample of incomplete current assignment behavior, as the majority of next-assignments were completed. When binning into completeness and incompleteness categories, the majority of data fell in the first two categories of completeness (3-4 problems solved 69%, 5-8 problems solved 21%), resulting in disproportionate pools for other categories (≤ 5%). These characteristics of the data made it more difficult to predict next assignment incompletion and instead, bias towards the larger current completeness categories. These models also only included measures of completeness (mastery speed and dropout rate) with a small selection of student behavior features. We started with simple models to identify the most logical predictive features. Moving forward, our analyses will include more holistic models of learning with parameters based on student and assignment features. Prior student knowledge and exposure, as well as problem content and difficulty, could be logical predictors of assignment completion and student learning. As such, future work will assess the generalizability of our three models while working to extend our predictive capacity further through the addition of new parameters.

We also plan to extend our models to predict next assignment performance. Similar to how we binned completeness to predict next assignment completeness, we can also use binary or binned correctness (partial credit) [15] to predict next assignment performance. This will expand our scope on higher level learning modeling and has the potential to provide more useful feedback to teachers when deciding on which content to review to increase the number of homework "finishers."

7. CONCLUSION

We have presented three predictive models of next assignment completion in ASSISTments that vary in complexity but perform comparably to one another. By modeling student performance at the assignment level, we were able to broadly model student behavior to predict whether students will be a homework "starter" or "finisher" on the next assignment. This approach to student modeling could serve as a foundation for a predictive teacher feedback tool within ASSISTments to increase teacher ability to target key content areas in class to increase the likelihood of all students being assignment finishers.

8. ACKNOWLEDGMENTS

We gratefully acknowledge funding from multiple NSF grants (1440753, 1252297, 1109483, 1316736, 1535428, 1031398), the U.S. Dept. of Ed. (R305A120125, R305C100024, P200A120238 and GAANN), and ONR.

9. REFERENCES

[1] Beck, J. E., and Woolf, B. P. 2000. High-level student modeling with machine learning. In *International Conference on Intelligent Tutoring Systems*. Springer, Berlin, Heidelberg, 584-593.

[2] Botelho, A. F., Adjei, S. A., and Heffernan, N. T. 2016. Modeling interactions across skills: A method to construct and compare models predicting the existence of skill relationships. In *Proceedings of the 9th International Conference on Educational Data Mining*, 292-297.

[3] Cen, H., Koedinger, K., and Junker, B. 2006. Learning factors analysis–a general method for cognitive model evaluation and improvement. In *International Conference on Intelligent Tutoring Systems*. Springer, Berlin, Heidelberg, 164-175.

[4] Corbett, A. T., Anderson, J. R., and O'Brien, A. T. 1995. Student modeling in the ACT programming tutor. *Cognitively diagnostic assessment*, 19-41.

[5] Duong, H.D., Zhu, L., Wang, Y., and 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*, 316-317.

[6] Harrison, A. 2018. EDMSubmission2018 Data. Retrieved March 7, 2018 from http://tiny.cc/EDMSubmission2018.

[7] He, J., Bailey, J., Rubinstein, B. I. P., and Zhang, R. 2015. Identifying at-risk students in massive open online courses. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[8] 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, 470-497.

[9] Pardos, Z. A. 2017. Big data in education and the models that love them. *Current Opinion in Behavioral Sciences*, 18, 107-113.

[10] Roll, I., Baker, R. S. J. d., Aleven, V., and Koedinger, K. R. 2014. On the benefits of seeking (and avoiding) help in online problem-solving environments. *The Journal of the Learning Sciences*, 23, 537-560.

[11] Romero, C., and Ventura, S. 2010. Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics-- Part C: Applications and Reviews*, 40, 6, 601-61

[12] Roschelle, J., Feng, M., Murphy, R. F., and Mason, C. A. 2016. Online mathematics homework increases student achievement. *AERA Open*, 2, 4, 1-12.

[13] Taylor, C., Veeramachaneni, K., and O'Reilly, U.-M. 2014. Likely to stop? Predicting stopout in massive open online courses. *arXiv preprint arXiv:1408.3382*, 273–275.

[14] VanLehn, K. 2011. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46, 4, 197-221.

[15] Wang, Y., Ostrow, K., Beck, J., and Heffernan, N. 2016. Enhancing the Efficiency and Reliability of Group Differentiation through Partial Credit. In *Proceedings of the 6th International Conference on Learning Analytics & Knowledge*. ACM, 454-458.

[16] Xing, W., Chen, X., Stein, J., and Marcinkowski, M. 2016. Temporal predication of dropouts in MOOCs: Reaching the low hanging fruit through stacking generalization. *Computers in Human Behavior*, 58, 119-129.

[17] Xiong, X., Li, S., and Beck, J. E. (2013, May). Will You Get It Right Next Week: Predict Delayed Performance in Enhanced ITS Mastery Cycle. In *FLAIRS Conference* (Vol. 2013).

FAQtor: Automatic FAQ generation using online forums

Ankita Bihani
Stanford University
Stanford, USA
ankitab@stanford.edu

Jeff Ullman
Stanford University
Stanford, USA
ullman@stanford.edu

Andreas Paepcke
Stanford University
Stanford, USA
paepcke@cs.stanford.edu

ABSTRACT

We explore implementation tradeoffs for automatically constructing a frequently asked questions list (FAQ) over multi-year class forum archives. Our ground truth was obtained from paid experts. They voted for or against inclusion of sample posts in a FAQ. Using the resulting labels we implemented and present two models for classifying posts for inclusion: logistic regression, and a random forest classifier. Results are presented for predicting both majority vote, and unanimous-yes decisions by the experts. We measured accuracy resulting from training on both, split dataset, and the full dataset with repeated 10-fold cross validation. When training on the full set the logistic regression classifier reaches an accuracy of 69% for predicting the majority decision, and 72% when predicting unanimous-yes. Random forest reaches 98% when using the full set. The predictors are easily obtained features from forum facilities, such as upvotes, unique views, and unique collaborations.

Keywords

Online Discussion forums, MOOCs, residential courses, FAQ, logistic regression, random forest, instructor support

1. INTRODUCTION

Online discussion forums empower instructors and students to engage one another in ways that promote critical thinking, collaborative problem solving, and knowledge construction [5] [7]. Several years' worth of a course forum archive hold treasures for a number of stakeholders. The answers to many relevant questions might be buried in those archives, and would save time for future students and teaching staff alike, if they were made available. Instructors could learn from those archives where students tend to falter, and modify their lectures accordingly. On the other hand, students could use the archive as a source for answers to questions without having to wait for responses.

Unfortunately, this potential is not tapped today. Questions are often re-raised, because their earlier answers are unavailable. This limitation motivates the construction of course specific frequently asked questions lists (FAQs). However, manual selection of archival posts for inclusion in a FAQ is not feasible. Thus, automated construction and maintenance of such a list is needed.

We present the details of FAQtor, an early version of our automatic FAQ generation system. Our contribution in this work is: (i) to show the initial prediction performance of logistic regression and random forest models in selecting forum posts for FAQs. And (ii) to show the relative importance of the readily available predictors that can be used for automatic FAQ selection.

2. AUTOMATIC FAQ GENERATION

We envision two types of FAQ lists, each for a different audience.

Students' FAQ : This FAQ is intended to include mostly conceptual questions. Given that the key concepts covered in most classes change very little over the years, this FAQ can help increase student productivity. For the course staff, the Students' FAQ translates to reduced work load in answering repeated questions.

Instructors' FAQ : This FAQ is meant to serve as a snapshot of the previous offerings of the course. The list is focused on topics that students struggled with in the past. The intent is for instructors to fix shortcomings in their courses, including operational mishaps. Entries in this FAQ thus need different types of entries than the list for students. The list should be useful as well when new instructors take over a course.

We focus here on generating the Students' FAQ.

The goal then is an algorithm that works well enough to pick initial posts for inclusion in frequently asked question lists. This task is not critical, so an approximate success strong enough to obviate the human selection of posts will suffice. The obvious choice for this task is a binary classifier, with a modest manually labeled set of posts serving as training and test sets.

We compared two classification mechanisms: logistic regression, and random forest. We present how both these methods performed.

3. OBTAINING A LABELED SET

We used a survey format in which the experts were presented with a series of Piazza [9] question–answer pairs, and notes. Experts were asked to make a binary *include/exclude* choice for each item.

The next section discusses how we chose contributions to be used in the survey, and created the ground truth.

3.1 Selecting Posts to Label

We started with the complete dataset of contributions on Piazza from four years of a graduate level Artificial Intelligence (AI) course. This course, offered by a large private university, contained an average of 1610 questions and notes per year from 2013 to 2016, with 2237 questions and notes in the most recent dataset. The ratio of the total number of questions and notes to the number of students on the forum was 11 in the most recent dataset. Given the large data volume we used simple heuristics to exclude questions and notes that would be irrelevant for the Student FAQ.

In our choice we considered the following features when deciding about including a post in the survey for the experts:

- Number of upvotes on the post (question or note)
- Number of upvotes on the students' answer (if the post was a question)
- Number of upvotes on the instructor's answer (if the post was a question)
- Number of unique collaborators in the follow-up thread of a candidate post
- Number of unique views of the post
- Length of the follow-up thread induced by a candidate post

We chose these features to ensure that our system was generalizable for use with other, similar forum datasets. These statistics are readily available or derivable in most forum facilities. Additionally, these choices were an easy way of indirectly having prediction features crowd-sourced. We did not at this point consider linguistic content analysis as a source for selection criteria.

Individual minimum threshold percentiles were set for each of the above features. These thresholds served to assemble the final post set for the expert opinion surveys. For instance, a threshold of 10th percentile on the number of question upvotes would mean that to be included in the survey, a question would need enough upvotes to clear that hurdle.

The thresholds for each of the above six features was set to a 5th percentile for the most recent class offering. As we progressively selected candidate FAQs from older datasets, these thresholds were gradually increased. Our assumption was that more recent offerings would have posts that are more relevant to upcoming offerings. Additional investigation is required to confirm this assumption. Our strategy was to keep the individual filters very low in order to cover the entire breadth of potentially interesting questions. However, only the questions that crossed the thresholds for *all* the features survived. We accumulated the candidate posts for inclusion in the survey using four years of forum data for the same course. Forty-one question-answer pairs and notes were then randomly sampled from this set.

Table 1: Low interrater reliability in all three groups

	Fleiss Kappa	p
G_1	-0.23	0.05
G_2	0.09	0.4
G_3	0.13	0.04

Thirteen additional posts were randomly sampled and included in the survey, to a total of 54. We allowed these additions to lie below the thresholds. We found strong support that the excluded posts were justifiably elided by observing the decisions of our experts. The experts selected none of the randomly sampled posts for inclusion in the FAQ. The posts were presented to the experts in random order.

The set of 54 items was partitioned into three batches. The batches were presented to three groups of three experts each: G_1, G_2, and G_3.

All the experts were recruited from among the current course instructors for the same class from which the forum contributions of the past few years were drawn. The survey instructions and one sample entry from the survey are included in the Appendix.

Table 1 shows that agreement around which posts are worthy of inclusion in a FAQ was low in all three expert groups. These values reflect that the experts' decision task was intrinsically subjective. We could not train the judges in how to make 'correct' decisions as we would in rating tasks with firm rules. Instead, we investigated two alternatives as ground truth: *majority vote*, and *unanimous yes.* The latter approach considers a post to be fit for a FAQ if all three judges agreed that the post is worthy of inclusion. We present prediction results for both notions of ground truth. Either choice provides a seed of posts for the FAQ list.

One approach towards continuous improvement after the initial list construction would be the inclusion of per-post voting by the student customers once the list is online. We could then add additional entries from the archives that are similar to upvoted items, or remove less useful contributions. We do not discuss this enrichment scheme here.

Our small set of human judgments was a challenge for both logistic regression (LR), and random forest (RF). We therefore compare in the results below two methods of applying each of these technologies. The first relies on the intrinsic randomization of 10-fold cross validation, repeated ten times for LR, and uses all 54 posts for training and performance estimation. In the RF this full-set method similarly relies on the randomness of RF feature and input selection, plus repeated CV, while using the full set of 54 posts.

The second method we investigated for LR and RF was the standard dataset split–70:30, which left us with 38 posts to

Table 2: Results for Logistic Regression

	LR-Split		LR-Full	
	Majority	**UnanYes**	**Majority**	**UnanYes**
Accuracy	0.56	0.63	0.69	0.72
Precision	0.6	1.0	0.9	0.83
Recall	0.38	0.25	0.36	0.43
F1	0.46	0.4	0.51	0.57

Table 3: Logistic regression confusion matrices for *UnanimousYes* and *majority* ground truths

	LR-Split UnanYes		LR-Full UnanYes	
	Exclude	Include	Exclude	Include
Exclude	8	6	29	13
Include	0	2	2	10

	LR-Split Majority		LR-Full Majority	
	Exclude	Include	Exclude	Include
Exclude	6	5	28	16
Include	2	3	1	9

train, and 16 to test. In this case both trainings were again run with repeated 10-fold CV.

3.2 Logistic Regression Classifier

After centering and scaling, we allowed the R *glmnet* training to find an optimal LASSO lambda via grid search. The optimal lamda was found to be 0.1. Prediction quality using the logistic regression classifier for both the data split (LR-Split), and use-all (LR-Full) methods are shown in Tables 2 and 3.

3.3 Random Forest Classifier

We trained our 4000-tree random forest classifier using 10-fold cross validation, repeated 10 times. The optimal *mtry* hyperparameter for the number of predictors to choose randomly while constructing trees was 2. The number 4000 of trees was determined empirically.

Figure 1 shows the decrease in GINI accuracy if each of the predictors were removed from use in the classification. The chart is sorted such that the highest predictor on the vertical axis is the most important, as it contributes most effectively to the decisions. Note that the low position of `no_upvotes_on_i_answer` and `no_upvotes_on_s_answer` is not entirely reliable. This effect stems from our candidate posts including both questions and notes. In Piazza, notes do not have instructors' nor students' answer options. Thus both, upvotes on student and instructor answers were set to 0 for notes, which was the most frequent value for these two measures in the other posts of our sample.

Notice the high placement of *unique views*. Views are the lowest-friction method for students to 'vote with their eyeballs', which then manifests strongly in the classifier.

Performance results for the 4K-tree random forest classifier are shown in Tables 4 and 5.

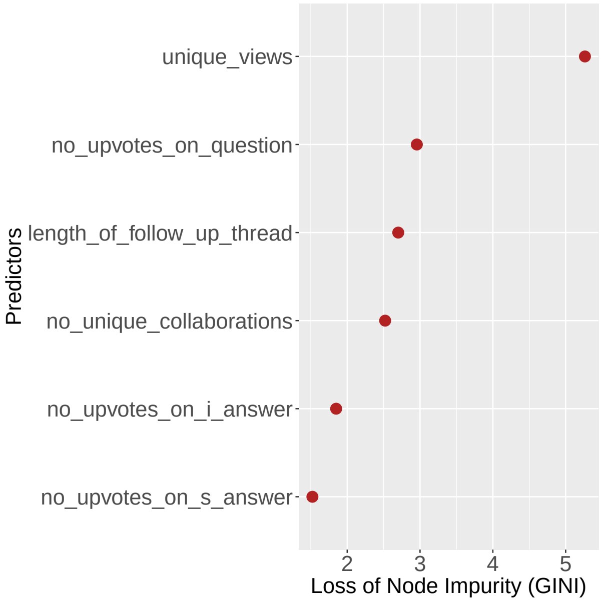


Figure 1: Decrease in GINI accuracy when removing specific predictors

Table 4: Results for Random Forest 4K

	RF4K-Split		RF4K-Full	
	Majority	**UnanYes**	**Majority**	**UnanYes**
Accuracy	0.5	0.63	0.98	0.98
Precision	0.5	0.75	1.0	0.96
Recall	0.25	0.37	0.96	1.0
F1	0.33	0.5	0.98	0.98

3.4 Discussion

We see that predicting *unanimous yes* leads to more reliable classification than predicting simple majority in all but one case, whether a split-set or full-set approach is used for training. Only for the random forest technology using the full data set are accuracies the same for predicting either outcome.

When comparing accuracies between split and full dataset approaches both logistic regression and random forest benefit from the full set. With more labeled training data we would expect this difference to narrow.

Whether predicting *unanimous yes* or *majority*, random forest reached higher accuracy than logistic regression. It is unclear whether the extremely high RF accuracy of 0.98 with the full set is overfitting in spite of random forests being known to resist this pitfall. Yet there is little doubt that RF is superior to LR for this data.

3.5 Deploying FAQtor in a Class Setting

We are exploring alternatives for allowing students to search and browse our forum archive. One supporting technology

Table 5: Random forest confusion matrices for *Unanimous Yes* and *Majority* ground truth

	RF4K-Split Majority		RF4K-Full Majority	
	Exclude	Include	Exclude	Include
Exclude	6	6	29	1
Include	2	2	0	24

	RF4K-Split UnanYes		RF4K-Full UnanYes	
	Exclude	Include	Exclude	Include
Exclude	7	5	30	0
Include	1	3	1	23

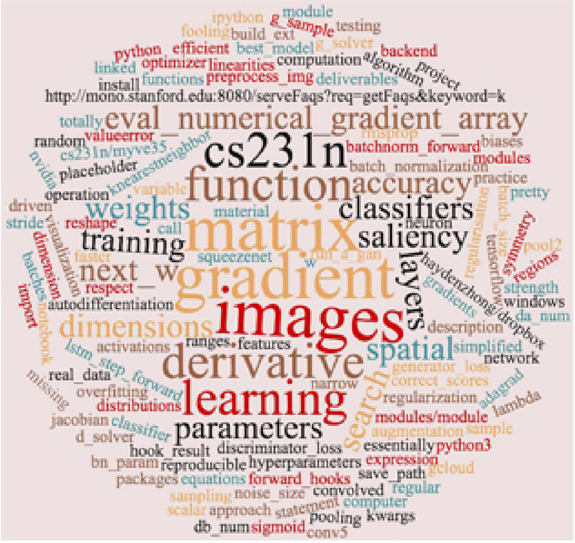


Figure 2: Example of clickable word cloud with keywords from the FAQ of a Convolutional Neural Networks for Visual Recognition class. Clicking on a word retrieves related posts from the archive.

towards a browsing interface is topic analysis of MOOC discussion content [3], which has been explored recently in [4] and [1].

For our browsing interface we use an adapted version of the Rapid Automatic Keyword extraction algorithm [8] to identify 'important' keywords in our FAQ. These keywords are then used to create an interactive word cloud, where clicking each keyword or topic leads the user to a list of question-answer pairs from the archive in descending order of relevance.(See Figure 2). We plan to deploy FAQtor in a real world class, and study usage logs, similar to the approach in [6]. By allowing students and instructors to vote entries in and out of the FAQ, we plan to capture patterns, and continuously improve the FAQ.

An important line of follow-on research needs to be an investigation into the generalizability of our results to non-science classes. The success of such transfer learning is not guaranteed, but is not out of the question [2].

4. CONCLUSIONS

We demonstrated how machine learning algorithms, in conjunction with available forum statistics, can reveal which question-answer pairs and notes are relevant and important for future offerings of a course.

Unlike much related work around frequently asked question lists, which assume the pre-existence of a knowledge base, we have sketched an approach for creating a solid starting point for such lists.

5. REFERENCES

[1] T. Atapattu and K. Falkner. A framework for topic generation and labeling from mooc discussions. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pages 201–204. ACM, 2016.

[2] A. Bihani and A. Paepcke. Quantyler: Apportioning credit for student forum participation. In *Proceedings of the 11th Conference on Educational Data Mining*, Buffalo, NY, US, 2018.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[4] A. Ezen-Can, K. E. Boyer, S. Kellogg, and S. Booth. Unsupervised modeling for understanding mooc discussion forums: a learning analytics approach. In *Proceedings of the fifth international conference on learning analytics and knowledge*, pages 146–150. ACM, 2015.

[5] R. M. Marra, J. L. Moore, and A. K. Klimczak. Content analysis of online discussion forums: A comparative analysis of protocols. *Educational Technology Research and Development*, 52(2):23, 2004.

[6] A. Moreo, M. Romero, J. Castro, and J. M. Zurita. Faqtory: A framework to provide high-quality faq retrieval systems. *Expert Systems with Applications*, 39(14):11525–11534, 2012.

[7] L. F. Pendry and J. Salvatore. Individual and social benefits of online discussion forums. *Computers in Human Behavior*, 50:211–220, 2015.

[8] P. rake 1.4.4. *Rake*.

[9] Wikipedia. *Piazza*, 2017.

APPENDIX

The survey instructions for ground truth collection were:
Please choose 'Yes' or 'No' for each of the following questions/notes.
A 'Yes' indicates that the respective forum item would be relevant and useful in a FAQ list for the class.
A 'No' means that the item should not be included in the FAQ for future offerings of the class.
The entries in the FAQ are taken from previous iterations of the class. Their intent is to answer student questions instantly, and reduce TA workload (by answering common student questions)

One sample item from the survey is as follows:
Question: *For the definition of Markov Blanket does it refer to finding the neighbors of A in terms of the factor graph or in terms of the Bayesian network*
Response: *The Markov blanket refers to all variables that have a factor in common with at least one of the variables in A. If two variables are connected in a Bayesian Network, then they must share a factor in the corresponding factor graph representation.*

Predicting if students will pursue a STEM career using School-Aggregated Data from their usage of an Intelligent Tutoring System

Jihed Makhlouf[1], Tsunenori Mine[1]
[1] Kyushu University, 744 Motooka Nishi-ku, Fukuoka, Japan
makhlouf_jihed@yahoo.fr, mine@ait.kyushu-u.ac.jp

ABSTRACT

It's been clear for years now that STEM (Science, Technology, Engineering, and Mathematics) fields workforce have a great impact worldwide. Efforts have been made to fill the gap between offer and demand in STEM positions and to encourage young students to enroll in STEM college major. However, enrolling in STEM major requires specific skills in maths and science that are taught earlier. Thus, interventions needed to be done in middle to high schools. Thanks to the increasing adoption of educational software, academic institutions have the possibility to gather fine-grained data about students usage of the software, and build predicting models using that data. For instance, researchers used data from an Intelligent Tutoring System (ITS) to predict student's STEM college enrollment [10]. In this paper we build a model that predicts if a student will pursue a career in STEM fields using data gathered from an ITS called ASSISTements. We propose a school-based approach where we aggregate students' features relatively to their peer school-mates. We compare this approach to a normal approach where no data transformation based on school is made. Our tests show a better AUC for the school-based approach attaining 0.601.

Keywords

STEM Career choice, Educational Data Mining, Predictive Analytics

1. INTRODUCTION

Science, Technology, Engineering, and Mathematics (STEM) fields are regarded worldwide as the building blocs for a nation's economy. Yet, STEM fields are facing shortage in manpower. Due to the importance of the problem, strategic decisions had to be made in order to find durable solutions. In fact when we think about filling the gap between the demand and supply in STEM fields position, it is hard to forget to mention the process of training such a high skilled manpower [6]. Several problems arise in college, when enrolled students find that they lack the necessary skills in maths and science and they face troubles when they reach the high level of mastery in college [4]. That causes a non-negligible number of students to drop out from the STEM major enrollment, which makes it more difficult to respond to the initial objective of filling the STEM fields positions. Consequently, a deeper analysis is required; one that goes into the early period in a student's academic journey, which is the middle school. It's in this period of time when students start to build their opinion and self-beliefs. It's also during that time where they are supposed to acquire the necessary skills that form the building blocks of their academic and professional life. Being able to detect students who face some troubles and providing the adequate support might be a decent solution. Yet, the traditional detectors rely heavily on students' grades and field observations, which are not helpful in detecting students that need help in the near term.

However, thanks to the advances in Information Technology, powerful educational software were developed and are rapidly being adopted by various academic institutions. They give the possibility to record finer-grained data about students activity within the software, opening the borders for more diverse and accurate models. Using data from the ASSISTments [1] ITS, researchers have built detectors of student knowledge, affects and behaviours [1, 2, 12]. Then proceeding by discovery with models, more predictive analysis were done, to predict the learning outcome, college enrollment [13] and more specifically STEM majors college enrollment [10].

In this paper, we aim at a longer term predictions of whether or not students will pursue a STEM career. We use data from the ASSISTments ITS. Our approach is to take students' performances and detectors' values and put them in context relative to their peer school-mates. We want to investigate if school-mates' data can improve model's predictions. To this end we measure the z-score for each student's features relative to his peer school-mates'. We compare it to the normal approach where data are not transformed in a school-based way. We also discuss which features of affects, performance and behaviours are good predictors, meanwhile, we introduce the usage of genetic programming in the process of finding the best machine learning pipeline for each approach.

2. METHODOLOGY

2.1 Data Acquisition

In order to proceed to our research, we used a large amount of data gathered from the ASSISTments platform. It's a web-based Intelligent Tutoring System provided by Worcester Polytechnic Institute free of charge. It targets middle school mathematics, where teachers can use a predefined set of content or they can create their own. The system provides students with the right assistance while assessing their knowledge. When students use the platform to work

[1] www.assistments.org

on problems assigned by their teachers, they receive immediate feedback whether they are correct or not. If they are right, they can proceed to the next problem, if not, the system provides them with scaffolding exercises which are subcomponents of the original problem to help students master the required skills to solve the problem.

The gathered data consists of actions log files representing click-stream interactions of students with the ASSISTments software. We count 942,816 actions stored in the log files coming from different types of student interactions, such as, requesting help, answering a question or even revealing a hint. Each action has a set of recorded information. Those actions were made by a group of 591 students, from 4 different schools, who used ASSISTments during the period between 2004-2007.

2.2 Features Exploration

The dataset contains 80 features; some of them were generated following a discovery with models approach; including student knowledge predictions, students behavioural features and affects. We also used other features that are directly related to the students' interactions with the software such as: number of problems solved within the system, time taken to answer a question, number of original and scaffolding problems, correctness in original and scaffolding problems, correctness in overall, number of hints used as well as bottom hint usage, and the number of help requests done as first attempts

2.3 Discovery with models

Several models have already been used to capture some of students behaviours or to predict their knowledge. In fact, for many years, predicting students knowledge was an active field of research [3, 9, 11, 5] that has shown the emergence of the Bayesian Knowledge Tracing (BKT) [3] as one of the most used models. Indeed, the BKT is able to estimate the student latent knowledge of a specific skill given previous observable performances.

Along with predicting the student's knowledge, different models were developed in order to estimate students' affects and disengaged behaviours. Researches such as [8] have produced 4 affective states detectors: Boredom, Engaged Concentration, Confusion, and Frustration. The disengaged behaviours appear in form of off-task attitude, gaming the system and carelessness.

2.4 Features Transformation and Selection

To make predictions related to the students' enrollment in a STEM career, we need to change the granularity of our data from the interaction level to the student level. Thus, we took the average of the selected features across all actions for each student. Picking the right features was done using the univariate feature selection, only keeping features that have strong relationship with the predicted variable (STEM job). Results of the selection process are shown in Table 1

Table 1: Univariate Features Selection

	STEM Career	Mean	Std	F-Value
Avg Bored	0	0.252	0.033	2.90e-05
	1	0.252	0.031	p=0.99
Avg Bottom hint	***0***	***0.046***	***0.035***	***10.811***
	1	***0.034***	***0.029***	***p<0.01***
Avg Carelessness	***0***	***0.12***	***0.065***	***18.207***
	1	***0.15***	***0.078***	***p<0.001***
Avg Confused	0	0.106	0.038	0.013
	1	0.105	0.035	p=0.910
Avg Correct Original	***0***	***0.43***	***0.156***	***11.458***
	1	***0.485***	***0.176***	***p<0.001***
Avg Correct Scaffold	***0***	***0.584***	***0.106***	***4.494***
	1	***0.606***	***0.101***	***p<0.05***
Avg Correct	***0***	***0.417***	***0.152***	***16.516***
	1	***0.471***	***0.144***	***p<0.001***
Avg Engaged Concentration	0	0.647	0.03	1.209
	1	0.650	0.026	p=0.271
Avg Frustration	0	0.127	0.047	1.834
	1	0.121	0.052	p=0.176
Avg FirstHelpRequest	0	0.285	0.066	1.126
	1	0.292	0.071	p=0.288
Avg Gaming	***0***	***0.113***	***0.124***	***4.115***
	1	***0.088***	***0.105***	***p<0.05***
Avg Hint	***0***	***0.266***	***0.141***	***14.108***
	1	***0.214***	***0.124***	***p<0.001***
Avg Knowledge	***0***	***0.224***	***0.135***	***16.881***
	1	***0.283***	***0.162***	***p<0.001***
Avg Off-Task	0	0.216	0.082	0.069
	1	0.219	0.074	p=0.792
Avg Original	***0***	***0.298***	***0.125***	***8.904***
	1	***0.337***	***0.139***	***p<0.01***
Avg Scaffold	0	0.418	0.114	0.573
	1	0.426	0.118	p=0.449
Avg Time Original	0	64.38	34.18	0.946
	1	67.82	38.16	p=0.331
Avg Time Scaffold	0	32.51	17.16	0.416
	1	33.64	17.99	p=0.518
Avg Time Taken	0	40.84	21.09	2.445
	1	44.25	23.51	p=0.118
Nb Problems	0	236.3	139.5	1.754
	1	255.1	143.9	p=0.185

After running the test we observed that only some features have a strong relationship with the predicted variable. In fact, correctness is a strong predictor not only in this study but also in previous studies interested in college enrolment [13, 10]. This is more emphasised when we look at the correctness in original problems, the difference in its mean value is higher compared to the difference of the mean value in correctness of scaffolding problems. It's due to the fact that scaffolding questions aim to help the students acquire the skill and help him solve the original problem. In a way, having higher correctness in original problems gives us more insight about the the student's skills. Another strong predictor is the average number of original problems, since it is the proportion of original problems over the total number of problems done by the student. Higher proportion of original problems translates to less "learning phase" through the scaffolding questions.

One of the interesting features is the hint functionality usage. Hints give the student some advices on how to solve a problem while explaining the skill. That's why students with high hint requests are more likely to pursue a non-STEM career. Furthermore, bottom hints explain the problem from it's basic notions. They are the lowest level of help, and that's why they are used less often but the difference between the two groups of students is still significant. Extensive hints usage has been reported as a detector for gaming the system behaviour [1], which is another strong predictor for students' enrollment in STEM career. Students who loose interests in STEM have higher mean values in gamin the system.

Additional features that can be good predictors are carelessness and knowledge estimation. Similarly to STEM major predictions [10], carelessness of students seems to increase when they are going to continue in a STEM career which is a non-intuitive finding shared by the two researches. Finally the average knowledge of a student is an estimation of his skills.

2.5 Approaches

Once the useful features are selected, we transformed the dataset to prepare for the first approach which considers the effect of the school on the student's career outcome. If we put the students' performance in the context of their surrounding, which, in this case, is the school, we might grasp some important information about the students' performances. So, the first approach, called school-based approach, is to separate students by their schools, then measuring the z-score of all students' features by school. This gives us a set of transformed data describing students relative to their peer school-mates. That was straightforward because all the students in the dataset had not changed their school while using ASSISTments. On the other hand, the normal approach is to simply use these features without distinguishing their school.

2.6 Optimization and genetic programming

Since we compare two different approaches independently, we want to find the most adequate machine learning method with its best hyper-parameters for each approach. We use genetic programming to find the best machine learning "pipeline" which is a combination of stacked machine learning techniques and their respective hyper-parameters. In fact, we do not compare two machine learning methods but rather try to give each approach its best shot.

Briefly, genetic programming is a technique derived from genetic algorithms in which instructions are encoded into a population of genes. The goal is to evolve this population using genetic algorithms operators to constantly update the population until a predefined condition is met. The most common ways of updating the population is to use two famous genetic operators called crossover and mutation.The population is evolving from one generation to another while keeping the fittest individuals in regard to one or many objectives. When using genetic programming for machine learning optimization, we use the pipeline score as the objective function. For example the pipeline accuracy score ca be considered as an objective function which has to be maximized.

In our case, we used genetic programming by searching through a multitude of machine learning techniques and their respective hyper-parameters, to find out which combination gives the best results. To achieve our goals we used the python library TPOT [7]. However, there are several genetic programming hyper-parameters that we need to initialize.

Table 2: Genetic Programming Hyper-parameters

Generations	Population size	Offspring size	Scoring
200	150	100	ROC AUC
Mutation rate	**Crossover rate**	**Internal Cross Validation**	
0.8	0.2	5 folds	

Table 2 explores the principal hyper-parameters that we have to initialize. The Generations count is the number of iteration of the whole optimization process. The Population size is the number of individuals which will evolve in each iteration. The Offspring size is the number of individual that is supposed to be generated from the previous population using the genetic algorithms' operators. After executing the operators and generating the offspring, individuals from the population and the offspring will compete to survive and be part of the next population (iteration i+1). Therefore we only keep the fittest ones, meaning the individuals with the best score. The method used to measure the score is the Area Under the Receiver Operating Characteristic Curve (ROC AUC). That means we only keep individuals (representing pipelines) which have the top values of ROC AUC. Mutation and Crossover rates are the probabilities of having respectively a Mutation or a Crossover operation to evolve one or more individuals. We set them to be 80% chance of having a mutation against 20% of having a crossover operation, which are common values. Finally, we proceed to an internal cross-validate for our pipelines, therefore we set the number of folds to 5.

We separately ran the optimization process for each approach and we ended up with two different machine learning techniques in two different pipelines. For the school-based approach we use a Gaussian Naive Bayes and for the normal approach we found that a Random Forest Classifier was the most efficient. Before running the optimization phase we did split the data into 2/3 for training (almost 400 students) and 1/3 for validation (almost 200 students) that we hold for the final validation at the end of the process. This split was stratified using the label (STEM Career) and the school, in order to respect the proportions of school diversity and STEM career outcome.

As shown in Table 3, the approach in which we z-scored the students features within their respective school gave us statistically significant better result than the normal approach with an AUC of 0.604 while the result of the normal approach is about 0.494. But in the case of RMSE, the normal approach had a better score of 0.425 compared to 0.476 achieved by the school-based approach.

Table 3: Validation score of both pipelines

	School-based	Normal approach
ROC AUC	0.604	0.494
RMSE	0.476	0.425

Once we know which methods and parameters to use, we proceed to training the cross-validated models using the whole dataset. But now, compared to the optimization phase where we had an internal 5-folds CV, we conducted a 10-fold stratified cross-validation training. And as previously, the folds were stratified in respect to the label (STEM Career) and the school id.

Table 4 shows the mean of the cross-validated values for both models. This time the school-aggregated model suffered an increase of RMSE which overpass 0.54 compared to its counter part. On the other hand the normal approach attained 0.521 in ROC AUC score but still lower than the score of the school model (0.601).

Table 4: Cross-validated scores for both approaches

	School-based	**Normal approach**
ROC AUC	0.601	0.521
RMSE	0.546	0.45

Even if the difference between the two approaches is statistically significant (p<0.01), the school-based approach has better AUC while the normal approach has lower RMSE, thus we cannot clearly confirm that the school-based approach has radically better results. The gain in terms of AUC is significant but it suffers from a relatively high RMSE.

3. DISCUSSION AND CONCLUSION

In this paper, we aimed at a longer term predictions of whether or not students will pursue a STEM career. Our approach was to take students performances and detectors values and put them in context relative to their peer schoolmates. We wanted to investigate if taking into account the local peer performances can improve model's predictions. Aggregating within the school gave us better ROC AUC scores but suffered from high RMSE suggesting that the improvement are not so big between both approaches. Perhaps, thinking about how well a student performs compared to his peers in the same school may not have a huge impact. But if we push the analysis further to aggregate student performances within his own classroom or to his teacher's students we can grasp some valuable informations of whether a teacher had an influence in the student's passion for STEM. Since it's the professor who is in contact with the students, it would be interesting to compare student's performances within a finer-grained entity which is the classroom.

4. ACKNOWLEDGMENTS

This work is partially supported by JSPS KAKENHI No. JP16H02926 and JP17H01843.

5. REFERENCES

[1] R. S. Baker. Modeling and understanding students' off-task behavior in intelligent tutoring systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 1059–1068, New York, NY, USA, 2007. ACM.

[2] R. S. Baker, A. T. Corbett, and K. R. Koedinger. Detecting student misuse of intelligent tutoring systems. In J. C. Lester, R. M. Vicari, and F. Paraguaçu, editors, *Intelligent Tutoring Systems*, pages 531–540, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[3] 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.

[4] D. F Whalen and M. Shelley. Academic success for stem and non-stem. 01 2010.

[5] J. Martin and K. VanLehn. Student assessment using bayesian nets. *International Journal of Human Computer Studies*, 42(6):575–591, 6 1995.

[6] R. Noonan. Stem jobs: 2017 update. Office of the Chief Economist, Economics and Statistics Administration, U.S. Department of Commerce(ESA Issue Brief 02-17), March 30 2017.

[7] R. S. Olson, R. J. Urbanowicz, P. C. Andrews, N. A. Lavender, L. C. Kidd, and J. H. Moore. *Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30 – April 1, 2016, Proceedings, Part I*, chapter Automating Biomedical Data Science Through Tree-Based Pipeline Optimization, pages 123–137. Springer International Publishing, 2016.

[8] Z. A. Pardos, R. S. J. D. Baker, M. O. C. Z. San Pedro, S. M. Gowda, and S. M. Gowda. Affective states and state tests: Investigating how affect throughout the school year predicts end of year learning outcomes. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, LAK '13, pages 117–124, New York, NY, USA, 2013. ACM.

[9] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis –a new alternative to knowledge tracing. In *Proceedings of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems That Care: From Knowledge Representation to Affective Modelling*, pages 531–538, Amsterdam, The Netherlands, The Netherlands, 2009. IOS Press.

[10] M. O. S. Pedro, J. Ocumpaugh, R. S. J. de Baker, and N. T. Heffernan. Predicting stem and non-stem college major enrollment from middle school interaction with mathematics educational software. In *EDM*, 2014.

[11] J. Reye. Student modelling based on belief networks. *Int. J. Artif. Intell. Ed.*, 14(1):63–96, Jan. 2004.

[12] M. O. C. Z. San Pedro, R. S. J. d. Baker, and M. M. T. Rodrigo. Detecting carelessness through contextual estimation of slip probabilities among students using an intelligent tutor for mathematics. In G. Biswas, S. Bull, J. Kay, and A. Mitrovic, editors, *Artificial Intelligence in Education*, pages 304–311, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[13] S. San Pedro, R. Baker, A. Bowers, and N. Heffernan. Predicting college enrollment from student interaction with an intelligent tutoring system in middle school. In *EDM*, pages 177–184, 01 2013.

Gamification and Student Engagement in an Online English Assessment System

Yu Yan
Penn State University
Keller Building
University Park, PA 16802
yzy122@psu.edu

Simon Hooper
Penn State University
Keller Building
University Park, PA 16802
sxh12@psu.edu

Shi Pu
Purdue University
Young Hall
West Lafayette, IN 47907
spu@purdue.edu

ABSTRACT

In this study, we investigated the effect of gamification on college students' engagement in an English reading-ability assessment system. Two versions of the assessment system were developed for the study: one version with gamification elements such as levels, a progress bar, and scores and the other version of the same mechanism but missing those gaming elements in the interface. A group of Chinese college students (N=342) were randomly assigned to use one of the systems during one semester (8 weeks). Preliminary results indicated that students using the gamification version stayed longer in the system during each session and spent more time in the system in total. In addition, the gamification effect was equally motivational for both male and female students.

Keywords

Gamification, Student Engagement, English online assessment, Gender

1. INTRODUCTION

Gamification is the use of game elements in a non-gaming context to improve user experience and motivation [5]. In a gaming environment, effectively designed gamification elements can create a sense of flow [4], resulting in improved concentration, joy, and involvement [10]. Results from many gamification studies suggest the positive influence gaming has on students motivation and performance [1–3, 6, 8].

However, the results of the effect of gamification are mixed in terms of gender and duration. Existing literature is far from reaching consensus on the role of gender in a gaming environment. For example, De Jean, Upitis, Koch, and Young [9] found that gender played a key role in learning outcomes and attitude in a gamified learning environment. However, another line of studies found that gamification had equal motivation on high school male and female students in a computer science course [14], and there was no gender effect on fifth grades math performance and attitude [12]. A recent literature review by Ke [11] summarized that gender may only influence game-play and learning processes rather than learning outcomes. Besides gender effect, little research has been done to investigate the long-term effect of gamification on students' behaviors and motivation. One previous research study argued that the increased engagement and interest brought by a gamified system may decay over time because this positive effect is due to a novelty effect [13]. Therefore, more research on the long-term effect of gamification and gender difference is needed.

This study investigated the use of a gamified formative assessment system with college-level Chinese English language learners. The study examined the effect of gamification on students' engagement across a semester and whether this effect is moderated by gender differences. The study results provide more evidence on the effect of gamification on students' engagement.

2. METHOD

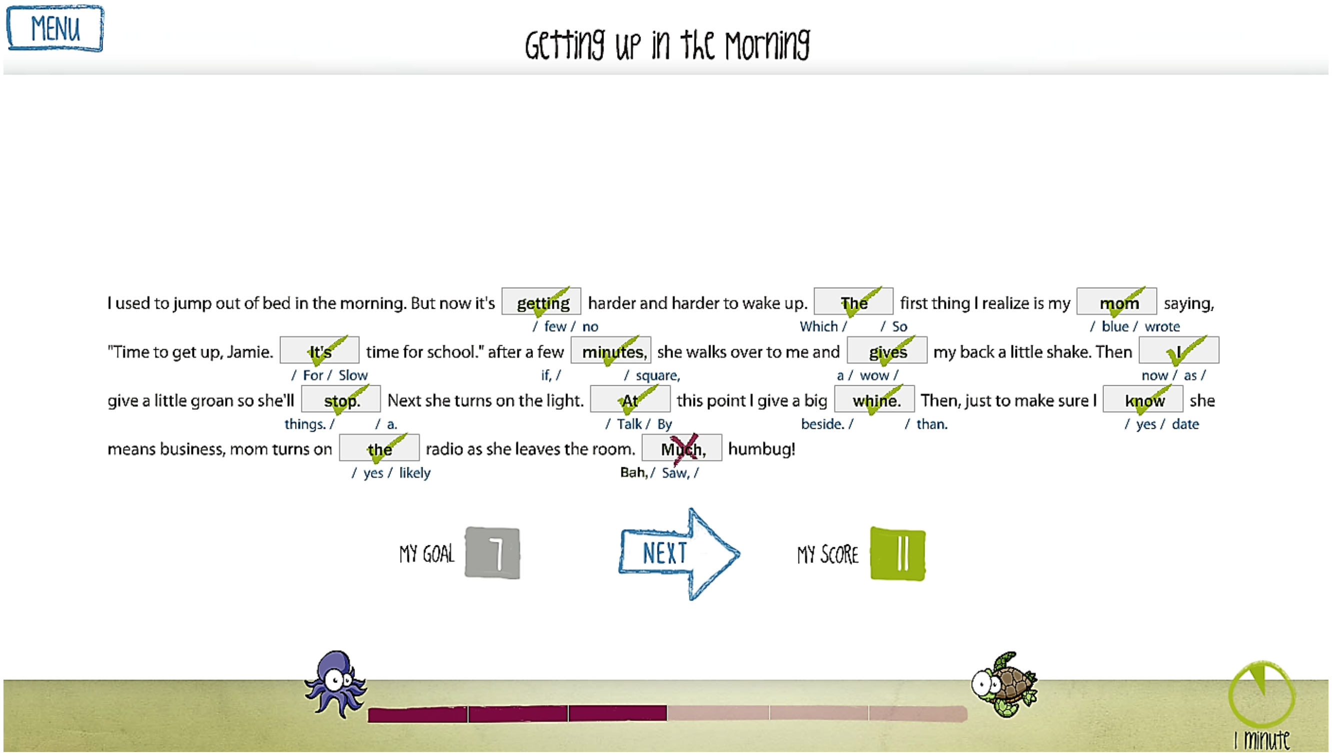
2.1 Participants and Settings

Participants were recruited from a four-year, second-tier regional college in Sichuan Province, southwest China. The college enrolls approximately 12,000 undergraduate students. All freshmen and sophomores are required to take a 90-minute English language class weekly each semester. Students from 14 classes taught by four English teachers (N=342) were recruited for this study.

2.2 Materials

The materials used in this study included two versions of the Maze tests generated by Avenue: PM, a web-based assessment system. A full description of the software (including the management system, different assessments and scoring rules) is presented elsewhere [7].

The Maze test used in the study is a cloze reading formative assessment. It involves presenting students with reading passages in which every seventh word is replaced with blanks. Students are given 60 seconds to complete each test. Two versions of the Maze were developed for this study. Both present students with Maze tests, but they differ in the presence or absence of gaming elements designed to enhance student motivation and two different interfaces. engagement. Figure 1 and Figure 2 demonstrate the screenshots of the two systems.



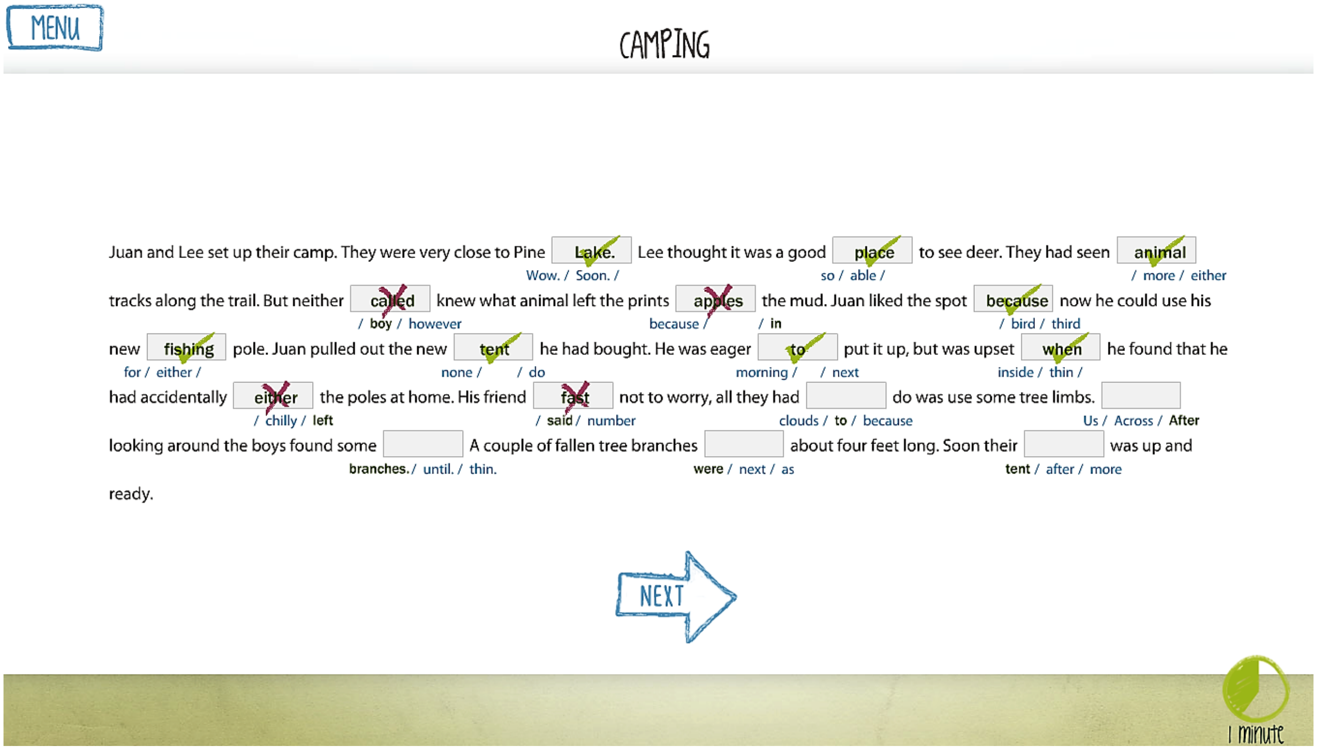
I used to jump out of bed in the morning. But now it's getting harder and harder to wake up. The first thing I realize is my mom saying,
/ few / no — Which / / So — / blue / wrote

"Time to get up, Jamie. It's time for school." after a few minutes, she walks over to me and gives my back a little shake. Then I
/ For / Slow — if, / / square, — a / wow / — now / as /

give a little groan so she'll stop. Next she turns on the light. At this point I give a big whine. Then, just to make sure I know she
things. / / a. — / Talk / By — beside. / / than. — / yes / date

means business, mom turns on the radio as she leaves the room. Much, humbug!
/ yes / likely — Bah, / Saw, /

Figure 1. Screenshot for Maze test with game features.



Juan and Lee set up their camp. They were very close to Pine Lake. Lee thought it was a good place to see deer. They had seen animal
Wow. / Soon. / — so / able / — / more / either

tracks along the trail. But neither called knew what animal left the prints apples the mud. Juan liked the spot because now he could use his
/ boy / however — because / / in — / bird / third

new fishing pole. Juan pulled out the new tent he had bought. He was eager to put it up, but was upset when he found that he
for / either / — none / / do — morning / / next — inside / thin /

had accidentally either the poles at home. His friend fast not to worry, all they had do was use some tree limbs.
/ chilly / left — / said / number — clouds / to / because — Us / Across / After

looking around the boys found some A couple of fallen tree branches about four feet long. Soon their was up and
branches. / until. / thin. — were / next / as — tent / after / more

ready.

Figure 2. Screenshot for Maze test with no game features.

2.2.1 Version 1: With gaming features

The gamified version of the Maze tests includes three elements to enhance students' motivation. The first element is a progress bar indicating the steps that must be completed to move up (or down) a level. The second element is the presence of visually appealing images of animal characters that represent the levels. Low levels use characters that are lower on the food chain (i.e., a jellyfish) and high levels use more advanced animals (i.e., an elephant). The third element is the display of goals to pass each level. The fourth element is the indication of progress towards a goal by comparing of user scores and test passing scores. Upon completing an assessment, students receive their score and information about whether they passed the current passage or not.

2.2.2 Version 2: Without gaming features

A version of the Maze was developed without the gaming features described above. Students do not know their current levels, their progress within a level, or their scores for each passage. Despite the interface differences between the two versions, the progress mechanism still operates. Thus, students are unaware of their progress although they may move forward or backward to different levels.

2.3 Procedure

Students were randomly assigned to one of the two groups, using Maze with or without gamifications. Students were told to complete tests using the software for around 10 minutes every week; however, they were free to leave the system anytime they wanted. Students' performances were captured and archived in a database for evaluation. Students received extra credit for participating the study.

2.4 Hypotheses and Data Analysis

The hypotheses of this study were:

Hypothesis 1: Students using a game-featured system will spend significantly **more time overall** in the system than the students who are using a non-game featured system.

Hypothesis 2: Students using a game-featured system will have significantly longer average time on system than the students who are using a non-game featured system.

Students' engagement was measured by the total and average time on system. Since the students were free to use and leave the system at any time, a longer time would indicate a higher level of engagement.

3. RESULTS

Table 1 demonstrates students' demographic information. 250 of the 342 students were female. Around 30% of students were male in both the control and the treatment group, which is consistent with the gender ratio across the college population.

Table 1. Students' grouping information

		without gamification		with gamification	
		N	percent	N	percent
gender	male	50	28.6%	42	25.1%
	female	125	71.4%	125	74.9%
total		175	51.2%	167	48.8%

3.1 Comparing the Total Time Spent

A two-sample t-test was used to examine this hypothesis. Table 2 demonstrates the descriptive data for total time of the two groups.

Table 2. Students' total time spend (in minutes)

	N	Mean	SD
without gamification	175	43.64	34.80
with gamification	167	50.95	36.72
total	342	47.21	35.88

Q-Q plot revealed that the total time was not normally distributed therefore a log transformation was performed on the total time before conducting the two-sample t-tests. After log transformation, the normality and homogeneity of variance assessed by Levene's Test were achieved. It found that students in the treatment group (using gamified version) spent significantly more time in the system during the study process than the students in the control group (using non-gamified version) (t = 2.02, p = .044).

In addition, gender was taken into consideration to exam whether gamification had an equal effect on the total time between different genders. Two-way ANOVA on gender and gamification showed that there was a main effect of gamification, F(1, 338) = 4.27, p = .0396, and a main effect of gender, F(1, 338) = 13.8, p = .0002. Female students spent 16.4 more minutes in total than male students. However, there was no interaction between gender and gamification, F(1, 338) = 0.65, p = .42, which indicate that the game features were equally effective for male and female students.

3.2 Comparing Average Time on System

Table 3 reports the descriptive data for the students' average time on system per visit, broken down by gamification and gender. Students using the gamification system spent 2.41 more minutes on average than students in the non-gamification group.

Log transformation was used before conducting the two-sample t-test to achieve the normality and equal variances. A t-test (t = 3.20, p = .002) showed that students spent significantly longer in the gaming system than students using the non-gaming system.

In addition, gender was taken into consideration to examine the effect of gamification on students' average session. Two-way ANOVA on gender and gamification indicated that there was a main effect of gamification, F(1, 338) = 9.93, p = .0018, and gender, F(1,338) = 5.30, p = .0219. Female students spent 1.34 more minutes for each session than male students. However, there was no interaction between gender and gamification, F(1, 338) = .69, p = .408, which indicates that the game features were equally effective for both male and female students.

Table 3. Descriptive information of average time on system (in minutes) by treatment groups and gender

	Without gamifications			With gamifications		
	N	Mean	SD	N	Mean	SD
Male	50	8.39	4.93	42	10.59	7.31
Female	125	9.53	4.07	125	11.95	8.14
Total	175	9.20	4.35	167	11.61	7.94

4. CONCLUSION

The purpose of this study was to investigate the effect of gamification (mainly on the progressive level and points) on college students' engagement while using an assessment system.

A preliminary finding of the study is that the gamification significantly improved college students' engagement in using an online assessment system. According to the results, students using the gamified system spent significantly more time in total and stayed longer for each session.[1] The results also demonstrated that gamification is effective for both male and female students, even though female students are significantly more engaged in using the system than male students in both conditions.

The current study has three particular strengths. First, gamification was not compared to a traditional paper-based assessment or a different computer-based assessment, but to an identical system except for a few gaming features in the interface. Therefore, the study provides convincing evidence for the studied game features (levels, progress bar, and scores on students' engagement. Secondly, recall that the participants were not kids, but college students who are less easily motivated by games. Arguably, the effect of the same gamified system on children or adolescents might be significantly larger than the effect we found in this study. The third strength is that we use engagement indicators derived from students' recorded behaviors in the system, which is more accurate and objective than subjective surveys. In addition, the engagement indicator can capture subconscious behaviors that are not even realized by the students.

The next step of data analysis would go further to understand the long-term effect of gamification on students. After looking at the total and average time on the system, we will investigate how the session length change over time (e.g. from week 1 to week 8). We would also investigate when students quit the system for each session. For example, did the students quit when they passed or failed a passage, or when they went up or went down a level? These would help us understand which gamification elements are more useful when motivating students.

5. ACKNOWLEDGMENTS

The authors would like to give thanks to the English instructors, Mr. Zhenggang Shi, Mr. Zhongjie Zhou, Ms. Haijing Shi, and Ms. Li Liu from Tianfu College of SWUFE, who helped to make this study possible.

[1] One alternative explanation for the longer time on system for the treatment group is that the UI of gamified system is significantly more complex than that in the non-gamified system. Therefore, students spent extra time to process the meaning of the features in the gamified system. We believe this does not explain the result for two reasons. First, before the start of the experiment, each student was given a user handbook corresponding to their system version. They were also given a practice test to familiarize themselves with the interface. Second, the gamification features are very straightforward, and the participants in this study are college students who should have the mental capacity to comprehend the gamification features reasonably fast. Thereby, we don't believe the time difference is a result of extra exploring time in the gamified system.

6. REFERENCES

[1] Attali, Y. and Arieli-Attali, M. 2015. Gamification in assessment: Do points affect test performance? *Computers & Education*. 83, (2015), 57–63. DOI:https://doi.org/10.1016/j.compedu.2014.12.012.

[2] Buckley, P. and Doyle, E. 2014. Gamification and student motivation. *Interactive Learning Environments*. (Oct. 2014), 1–14. DOI:https://doi.org/10.1080/10494820.2014.964263.

[3] Charles, D. et al. 2011. Game-based feedback for educational multi-user virtual environments. *British Journal of Educational Technology*. 42, 4 (2011), 638–654. DOI:https://doi.org/10.1111/j.1467-8535.2010.01068.x.

[4] Csikszentmihalyi, M. 1990. *Flow: The psychology of optimal experience*. Harper & Row.

[5] Deterding, S. et al. 2011. Gamification: Using game design elements in non-gaming contexts. *CHI '11 Extended Abstracts on Human Factors in Computing Systems* (New York, 2011), 2425–2428.

[6] Hamari, J. et al. 2014. Does gamification work? A literature review of empirical studies on gamification. *2014 47th Hawaii International Conference on System Science* (Jan. 2014), 3025–3034.

[7] Hooper, S. et al. 2013. Considering the design of an electronic progress-monitoring system. *Handbook on Design in Educational Computing*. R. Luckin et al., eds. Routledge.

[8] Jackson, G.T. and McNamara, D.S. 2013. Motivation and performance in a game-based intelligent tutoring system. *Journal of Educational Psychology*. 105, 4 (2013), 1036–1049. DOI:https://doi.org/10.1037/a0032580.

[9] De Jean, J. et al. 1999. The Story of Phoenix Quest: How girls respond to a prototype language and mathematics computer game. *Gender and Education*. 11, 2 (1999), 207–223. DOI:https://doi.org/10.1080/09540259920708.

[10] Johnson, D. and Wiles, J. 2003. Effective affective user interface design in games. *Ergonomics*. 46, 13–14 (2003), 1332–1345. DOI:https://doi.org/10.1080/00140130310001610865.

[11] Ke, F. 2009. A Qualitative Meta-Analysis of Computer Games as Learning Tools. *Handbook of Research on Effective Electronic Gaming in Education (3 Volumes)*. 1759.

[12] Ke, F. and Grabowski, B. 2007. Gameplaying for maths learning: Cooperative or not? *British Journal of Educational Technology*. 38, 2 (2007), 249–259. DOI:https://doi.org/10.1111/j.1467-8535.2006.00593.x.

[13] Koivisto, J. and Hamari, J. 2014. Demographic differences in perceived benefits from gamification. *Computers in Human Behavior*. 35, (2014), 179–188. DOI:https://doi.org/10.1016/j.chb.2014.03.007.

[14] Papastergiou, M. 2009. Digital game-based learning in high school Computer Science education: Impact on educational effectiveness and student motivation. *Computers & Education*. 52, (2009), 1–12. DOI:https://doi.org/10.1016/j.compedu.2008.06.004.

A first attempt to address the problem of overbooking study programs

Karin Hartl
University of Applied Sciences Neu-Ulm
Wileystr. 1, 89231 Neu-Ulm/Germany
+49 (0) 731-9762-1531
karin.hartl@hs-neu-ulm.de

1. INTRODUCTION

The demands made on universities in Germany have significantly changed in recent decades, transforming them into service providers. The student, who is the main customer of a university, has an enormous variety of "product" choices. These products include all different kinds of degree programs, from full-time to part-time programs, international study offers and online courses. In addition, the student can choose from various kinds of Higher Education Institutions (HEI) – e.g., universities, business schools and digital universities – and from HEI institutions all around the world. Therefore, universities need to develop their professional management and constantly increase their reputation to ensure long-term success and existence.

A major challenge for universities that significantly impacts their success and reputation is the overbooking or underbooking of study programs. Overbooking is a phenomenon that happens when more units from a limited capacity are sold than are actually available [1]. In the airline and hotel industry, this is happening to avoid revenue loss from no-show customers [2, 3]. Without overbooking, every cancellation would lead to an empty room or seat and consequently to a loss in income [1, 4]. However, if every booking does appear, there might not be enough available spaces for each customer, meaning the hotel or airline face additional costs, as they must provide customers with lucrative alternatives to give-up their booking. Otherwise, customer dissatisfaction is high, which would most likely result in bad reviews and a low degree of esteem.

Universities face the same problem. In Germany, potential students have an enormous choice of universities and study programs. In general, they apply for more than one program and to several different universities. As a result, only a relatively small number of people applying to a degree program are actually willing to start the program once accepted. As universities in Germany get a portion of their government funding for their first semester student numbers, the capacity utilization is very important. In addition, existing resources – human resources, assets, equipment – are not exploited to their full potential; however, they still must be available and subsidized. Therefore, universities regularly overbook their available study spaces and hope that in the end exactly the right number of applicants will matriculate.

Conversely, if course capacities are overbooked, the available resources are overloaded. The lecture theaters are overcrowded, lecturers are overworked, and supplementary services are working to their limits. This situation is visible to all, staff members, students and external stakeholders, and will most likely lead to dissatisfaction on all sides.

The decision on the exact number of applicants invited to the program is, in practice, widely based on the experience of the professionals in the admissions department; it is accordingly often instinctive and based on the experiences of previous semesters. Further information is necessary to assure an accurate estimation of no-show applicants. Consequently, universities should use all the resources available to support the concerning decision-making process. During the application process, each university collects data about the applicants and potential students. We assume that the collected data contains supporting information. With Data Mining techniques, universities have the opportunity to extract information from existing data resources and forecast the no-show of applicants, which can objectively support the relevant decision-making process and positively influence the long-term success and existence of a university.

This research presents a first attempt at calculating a prediction model that can help to forecast the no-show of students. Therefore, we analyzed data from the application period for the winter semester of 2017/18 for a small German university, using decision tree modelling, rule induction and logistic regression analysis.

2. ANALYSIS

2.1 Approach

For analysis purposes, we extracted applicant data from the application period for the winter semester 2017/18. The dataset contains data of six different bachelor study programs, namely business administration, business administration in health, information management, information management automotive, industrial engineering and industrial engineering in logistics. In total, the dataset is comprised of 25 attributes and 1,830 examples. The attributes that are considered interesting for analysis purposes are presented in Table 1. The final admission status (*AS*) was extracted from the system around six weeks before the official beginning of the winter semester. The final matriculation status (*Status*), which is the target variable for the predictive analysis, was extracted four weeks after the official beginning of the semester.

2.2 Descriptive Analysis

In Figure 1, the deviation of the applicants across the study programs is illustrated. The majority of applicants applied for the business administration program (29.8%) and the study program with the least number of applicants is industrial engineering logistics (9.8%). In terms of *Priority*, 86% of students chose the first-preference subject and 11.2% their second-preference subject. Only 2.8% had made an application to a program which did not appear in their first two preferences.

The average applicant at our case university is 21 years old, and we have in total 909 male and 921 female applicants. From these applicants, 243 females and 206 males enrolled to one of the programs as students at the beginning of the semester. There are in total 449 students, 24% of the whole dataset. Accordingly, 76% of the students that did apply for one of the study programs until 6 weeks before the start of semester did not matriculate at the university.

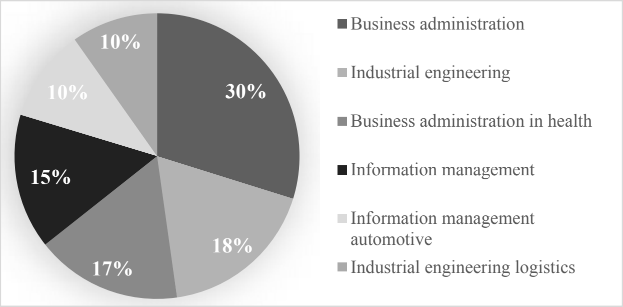
Table 1. Attributes in the dataset.

Attribute	*Description*
ID	The individual student ID identifies each example and ensures the anonymity of the applicants in the dataset.
Date of Birth	The exact date of birth of the applicant, which is used to calculate the age.
Place of Birth	The birthplace of the applicant.
Age	The age of the applicant at the time of the application.
Gender	The gender of the applicant that is either f = female or m = male.
Matriculation (Status)	An attribute that indicates if the applicant enrolled at the beginning of the semester (true) or if she/he did not enroll (false).
Place	Name of the place, where the applicant resides at the time of application.
Postcode (PC)	The postcode of the place of residence.
Nationality (Nat)	The nationality of the applicant.
GerNat	This attribute tells us if an applicant is of German nationality (*Yes*) or is not (*No*).
HEEQ type	The kind of Higher Education Entrance Qualification (HEEQ) that the student obtained to qualify for study. An overview of the various kinds can be found at Hochschulstart [5].
HEEQ grade	The grade of the HEEQ certificate. In Germany this can be 1= excellent, 2 = Good, 3 = satisfactory, 4 = sufficient and all the decimals numbers in between.
First Semester (FS)	This attribute describes if a student starts her/his first semester at university (Yes) or if she/he studied before in another program or university. This attribute is important, as German universities get funding for the students which start their first university semester (FS) in one of their programs.
Number of previous semesters (PS)	If the applicant already has FS, this attribute shows the number of semesters she/he has already studied at this or another university.
Admission Status (AS)	During the application process, each applicant passes through various stages, which are indicated with the admission status. The status can either be *admitted*, *received university place offer*, *place rejected* (by the applicant) or *place accepted.* The status *admitted* means that the student fulfills the necessary requirements to be admitted to the program. This is proven by documents which must be provided and sent by the student. Afterwards, the student *receives* a study place offer, which he/she can either *accept* or *reject*.
Priority	This attribute indicated the priority of a student for a specific study program. It can be between 1 and 6, with 1 indicating the highest priority.

In terms of study experience, 76.6% of the applicants had not previously studied at university and 23.4% have previous study experience. The students with study experience had, on average, studied for three semesters at this or another university.

The attributes that give some information about the history of the applicants is *HEEQ kind* and *HEEQ grade*. Figure 2 illustrates the student deviation according to their HEEQ and their gender, and Table 2 illustrates the *HEEQ grades* of the applicants in correspondence to the main *HEEQ kinds*. The HEEQ certificate can be achieved in several ways. Classically, students attend the *Gymnasium* and graduate after successfully finishing the 12th or 13th grade. This certificate qualifies the students to study at the HEI of their choice. If students do not attend *Gymnasium*, they can obtain their qualification for higher education through continuous education. The most common forms are the *Fachoberschule*, the *Berufsoberschule* or a *Kolleg* [6]. In the *Fachoberschule*, students gain a topic-related HEEQ. The *Berufsoberschule* is for students with a finished apprenticeship that want to qualify for higher studies, and the *Kolleg* can either offer a way to make the HEEQ in the evening, full-time or topic related.

Figure 1. Deviation of the applicants in the data dataset.



The main body of HEEQ of our applicants attended the *Gymnasium*. Furthermore, we see that a notable number of male applicants attended the *Berufsoberschule* or the *Kolleg*. Accordingly, our male applicants often have practical experience and a finished apprenticeship, whilst female applicants are more likely to directly apply to university after finishing school.

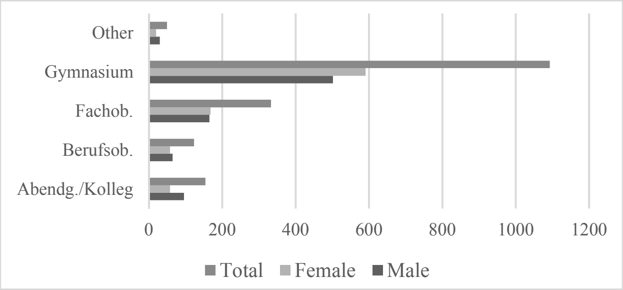


Figure 2. Deviation of applicants according to *HEEQ kind*.

Table 2. *HEEQ grades* in correspondence to the main *HEEQ kinds*.

	Abend-gymnasium	Fachober-schule	Gymnasium	Berufs-obersch.	Other
< 1.9	25	21	127	12	11
2.0	8	5	38	7	11
2.1	9	8	59	6	8
2.2	5	6	51	10	11
2.3	10	17	62	13	7
2.4	1	6	97	5	3
2.5	11	27	117	1	6
2.6	7	29	90	9	7
2.7	12	16	81	15	18
2.8	13	29	65	3	12
2.9	7	22	69	9	0
3.0	21	33	54	5	8
> 3.0	25	114	182	28	26
Total	154	333	1092	123	128

2.3 Data Mining Analysis

2.3.1 Data Preparation

The goal attribute of our analysis is the *Status* attribute, which tells us if an applicant actually matriculated (*true*) and started as a student in the study program. Of our remaining dataset, only 357 cases have in fact started as students at the university and the remaining 976 cases did not matriculate. Therefore, our dataset is slightly skewed. Skewed data can be addressed in various ways. We chose to balance our dataset at this stage of the analysis through only considering 400 of the applicants' cases who did not start at the university (*Status = false*). Hence, a final number of 757 examples – 400 students that did not enroll and 357 students that did enroll – remained for the following analysis.

2.3.2 Decision tree

First, the data has been analyzed with a decision tree modeler. Several decision tree algorithms are available in *RapidMiner*. The standard decision tree operator is a combination of various algorithms and is able to consider all data types for analysis [6]. It is robust to missing values and has a number of pruning options which are easy to adjust by the analyzer. The generation of the tree is based on the concept of information entropy that splits the dataset according to a measure of the maximized information gain and the reduction in entropy [7]. The information gain obtained towards the label attribute by splitting the data at a specific attribute B can be described as:

$$Gain(B) = Info(E) - Info_B(E)$$

where *Info(E)* is the original information requirement and $Info_B(E)$ is the new information requirement after partitioning the dataset on a specific attribute B[7].

Several decision trees have been calculated, adjusting the pruning parameters and comparing the performance measure predictive accuracy. The pruning parameters of the final decision tree are described in Table 3, and the tree itself is described in Figure 3. The generated tree shows a good overall accuracy (Table 4), meaning that the model predicted the correct matriculation status (*Status*) in 88% of the cases. The class recall for both target classes – *true* (student matriculates) and *false* (potential student does not matriculate) – is of high value too and the same can be said for the class precision. If we focus on the target class *true*, then we can see that our model is able to identify 85% of the potential students who will matriculate at the beginning of the semester.

For the applicants that did not enroll at the beginning of the semester, this value is even higher, and in 90.5% of the cases, the decision tree was able to predict the student no-show.

The applicants who had accepted the study place offer by the six-week date before the official beginning of the semester are more likely to matriculate to the study program of their choice, if they do not have previous study experience and are between 17 and 19 years of age. Furthermore, students that are between 20 and 22 years of age, with a very good HEEQ grade (equal to or above 1.85), and no previous study experience are also likely to attend the program. According to the model this is also true for applicants with no previous study experience, an age between 20 and 22, and a HEEQ grade below 3, as well as for female students between the age of 20 and 22, with a HEEQ grade higher than 2.85, and no previous study experience.

Table 3. Final pruning parameter for decision tree model.

Parameter	Setting	Description
Confidence	0.25	This is the confidence level used for pessimistic error calculation.
Minimal gain	0.02	The tree nodes are split only if they are greater than the pre-defined minimum gain. The higher the min. gain, the viewer splits will be in the decision tree.
Minimal leave size	10	The minimum number of data objects that form a leave node.
Minimal size of split	20	The minimum number of examples that need to be in a tree node for it to be split on.

Figure 3. Description of final decision tree model.

```
AS = Received place offer {false=138, true=31}
AS = place accepted
|   FS = false
|   |   Age = 17-19: true {false=8, true=148}
|   |   Age = 20-22
|   |   |   HEEQ grade > 1.850
|   |   |   |   HEEQ grade > 3: true {false=4, true=46}
|   |   |   |   HEEQ grade ≤ 3
|   |   |   |   |   Gender = M: true {false=3, true=33}
|   |   |   |   |   Gender = W
|   |   |   |   |   |   HEEQ grade > 2.85: false {false=6,
                                                      true=6}
|   |   |   |   |   |   HEEQ grade ≤ 2.85: true {false=9,
                                                      true=38}
|   |   |   HEEQ grade ≤ 1.85: true {false=0, true=11}
|   |   Age = 23-25: true {false=6, true=33}
|   |   Age = 26 and older: true {false=0, true=10}
|   FS = true
|   |   HEEQ grade > 2.95: false {false=20, true=0}
|   |   HEEQ grade ≤ 2.95
|   |   |   Gender = M: false {false=13, true=4}
|   |   |   Gender = W: true {false=5, true=6}
AS = place rejected: false {false=72, true=8}
AS = admitted: false {false=116, true=6}
```

Applicants that are, six weeks before the official start of the semester, still not advanced in the application process and have only been admitted by this time, are most probably not attending the program. Furthermore, applicants that accepted the study place offer and have had previous study experience have, according to the model, a high probability of not actually starting the studies at the official beginning of the semester.

Table 4. Confusion matrix for the final decision tree model.

Accuracy: 88.08 +/- 5%			
	True false	True true	**Class precision**
Pred. false	362	55	86.81%
Pred. true	38	325	89.53%
Class recall	90.50%	85.53%	

2.3.3 Rule Induction

With the rule induction Data Mining process IF-THEN rules are deducted from a dataset [8]. The *RapidMiner* rule induction approach works with the Repeated Incremental Pruning to Produce Error Reduction (RIPPER) by Cohen [9]. The algorithm builds rules one by one and focuses on one class label first, before moving on to the next class, which in our example is *Status (true)* or *Status (false)*. The rules r_i are built by adding conjunctions one by one to an empty rule set with the aim of increasing the overall rule accuracy [8].

For our analysis, we split the dataset into a training set (80% of the dataset) and a testing set (20%). Afterwards, we analyzed the dataset several times, including and excluding attributes. The results have been compared according to the performance accuracy of the rules, which are presented in Table 5.

In Scenario 1, we included all the criteria of Table 1, before we iteratively removed attributes to improve the performance. In Scenario 2, we removed all the attributes from the analysis that were additionally represented through other attributes. These attributes are *Date of Birth* (*Age*), *Postcode* (*Place of Residence*), *Nationality* (*NatGer*) and *Number of Previous Semesters* (*First Semester)*. In Scenario 3, the *Place of Birth* attribute was removed from the analysis as well, because the descriptive analysis indicated that the main body of our applicants had been born in the region. Accordingly, we assume that this attribute does not contribute to classifying the dataset. The same is applicable for the attribute *Place*, which has additionally been removed in Scenario 4.

The results indicate that Scenario 4 is the best possible solution for the rules in our dataset, with an overall model accuracy of 79.49%. As this is significantly lower than the performance of the decision tree, the rule induction results are not considered for interpretation at this stage of the analysis.

Table 5. Performance measures of the rule induction scenarios.

Scenario	Model Accuracy
Scenario 1	60.90%
Scenario 2	60.90%
Scenario 3	68.59%,
Scenario 4	79.49%

2.3.4 Binominal logistic regression

Another suitable approach to model if an applicant enrolls at the beginning of the semester is the binominal logistic regression, because our target attribute is binominal. The logistic regression model calculates the probability of an event happening, which in our case is either *Status (true)* or *Status (false)*. Conforming to the above analysis approaches, we calculate several logistic regression models. The different models present the above described scenarios 1 to 4 and are again compared through the overall model accuracy (see Table 6).

Table 5. Performance comparison of the logistic regression models.

Scenario	Model Accuracy
Scenario 1	77.82%
Scenario 2	65.26%
Scenario 3	64.62%,
Scenario 4	87.69%

The best solution has been achieved with the settings of scenario 4, which only including *Age*, *Gender*, *Status*, *GerNat*, *HEEQ type*, *HEEQ grade*, *FS*, *AS* and *Priority* in the analysis, with an overall model accuracy of 87.69%.

The coefficients of the regression model indicate whether an attribute has a positive or a negative influence on the probability of a student matriculating. The model only shows a limited number of significant attributes (p-value < 0.05), which is also represented in a relatively low determination coefficient (R^2 = 0.618). Nevertheless, the results indicate that if the student has previous study experience (*FS*=*true*) and the *AS* is either *admitted* or *rejected study place offer,* the probability decreases that she/he will matriculate at the beginning of the semester (see Table 7). Furthermore, the *AS* status of *accept study place* as well as a *HEEQ grade* of 2 or 2.1 positively influence the probability of a student starting the study program.

Table 7. Significant attributes in the logistic regression model.

Attribute	Coefficient	p-value
AS = accepted place offer	3.55	0.000
AS = admitted	-1.51	0.002
Age = 26 or older	1.83	0.007
HEEQ grade = 2.2	2.02	0.014
AS = rejected place offer	-1.1	0.017
HEEQ grade = 2.1	1.44	0.048

3. RESULTS AND NEXT STEPS

This first attempt to get more insight into the challenge of overbooking study programs at German universities by using the available applicant data resources shows promising results. The decision tree model and the logistics regression model are both able to predict the no-show of students with an accuracy of approximately 90%. Therefore, the university admissions departments can use the information provided by these models to plan further courses of action and ensure that the study spaces are used to full capacity at the beginning of the semester.

For example, if many applicants have not accepted their admittance or study place offer six weeks prior to the beginning of the semester, the university needs to find either more applicants or to actively address the existing applicants and win them as students. According to the information provided by the decision tree, they could focus on applicants with no previous study experience. Both models show, if students with no previous study experience accept the place offer, they are highly likely to start as students at the beginning of the semester.

Besides such interesting first insights which the decision tree and logistic regression model provide, the research is still in its early stage and not without limitations. Only a low number of attributes have been identified as informative by the model, providing limited insights to the decision makers. This can be connected to the relatively small dataset on which the results are based. Furthermore, no practical implementation and testing of the model has been done so far.

As a result, we have collected data from the application period of the summer semester 2018 and include this data to re-assess the results. Afterwards, we plan on applying the model on the applicant data of the winter semester of 2018/2019 and on testing its practical usefulness.

4. REFERENCES

[1] Zenkert, D. 2017. No-show Forecast Using Passenger Booking Data. Lund University, 2017.

[2] Hueglin, C. and Vannotti, F. 2009. Data mining techniques to improve forecast accuracy in airline businesses. In Proceedings of the ACM SIGKDD International Conference in Knowledge Discovery and Data Mining (2009), San Francisco.

[3] Klindokmai, S., Neech, P., Wu, Y., Ojiako, U., Chipulu, M. and Marshall, A. 2014. Evaluation of forecasting models for air cargo. The International Journal of Logistics Management. 25, 3 (2014).

[4] Phumchusri, N. and Meneesophon, P. 2014. Optimal overbooking decision for hotel rooms revenue management. Journal of Hospitality and Tourism Technology. 5, 3 (2014).

[5] Hochschulstart Arten der Hochschulzugangsberechtigung. 2018. URL = https://www.hochschulstart.de/index.php?id=hil fe501_arthzb. Hochschulstart Arten der Hochschulzugangsberechtigung.

[6] RapidMiner Decision Tree. RapidMiner, 2018. URL = https://docs.rapidminer.com/studio/operators/modeling/predictive/trees/parallel_decision_tree.html.

[7] Han, J., Kamber, M. and Pei, J. 2012 Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, Waltham, USA.

[8] Kotu, V. and Deshpande, B. 2015. Predictive Analytics and Data Mining. Morgan Kaufmann by Elsevier Inc., USA.

[9] Cohen, W. W. 1995. Fast Effective Rule Induction. International Conference on Machine Learning. 12 (1995).

Predicting Student Performance: The Case of Combining Knowledge Tracing and Collaborative Filtering

Solmaz Abdi
The University of Queensland
solmaz.abdi@uq.edu.au

Hassan Khosravi
The University of Queensland
h.khosravi@uq.edu.au

Shazia Sadiq
The University of Queensland
shazia@itee.uq.edu.au

ABSTRACT

In the past few years, many competing learning models have been proposed for improving the accuracy of predicting student performance (PSP). A well-studied subclass of algorithms focused on PSP uses temporal models to determine the knowledge state of users. Bayesian Knowledge Tracing (BKT), as one of the leading models in this subclass, uses Hidden Markov Models to capture the student knowledge states. An emerging new subclass of algorithms focused on PSP uses collaborative filtering, which is used primarily by recommender systems. Matrix Factorization (MF), a leading model in this subclass, can be presented as a rating prediction problem where students, tasks, and performance information are mapped to users, items and ratings, respectively. BKT and MF complement each other's strengths and limitations quite effectively. In particular, BKT relies on four skill-specific parameters for learning the sequential behavior of learners on each concept, but it does not capture the similarities among users and items. In contrast, MF uses latent factors to exploit the similarities among users and items from learner-item performance, but disregards any temporal effect in modeling student learning. In this paper, we aim to investigate the effect of combining variations of BKT and MF using a proposed algorithm that exploits the power of MF in modeling the implicit similarities among learners and items while using the explicit parametrization of BKT towards improving PSP. Our results on four benchmark educational datasets show that our approach outperforms the base classes as well as traditional techniques such as linear regression, logistic regression and Neural Networks for combining BKT and MF.

1. INTRODUCTION

Heavily studied in the community of educational data mining (EDM), the problem of predicting student performance (PSP) uses observations from students' behavior to find a model that predicts their future performance on unseen learning tasks [3].

Temporal models have been used extensively for PSP and determining the knowledge state of users. They rely on the sequential behavior of learners to model their learning. In these models, the students' performance on the next task is predicted using their performance on their prior test items [11] and a Q-matrix [1], which is a binary matrix that shows the relationship between test items and underlying concepts. One of the leading temporal models for PSP is Bayesian Knowledge Tracing (BKT) [3]. BKT uses Hidden Markov Models for capturing students' knowledge states as a set of binary variables. While BKT has received significant attention and improvement since it was first proposed, it is unable to capture similarities among learners or items, which has shown to be an important aspect in improving PSP [14].

Applying collaborative filtering (CF) techniques is another promising approach for PSP. One of the most successful collaborative filtering techniques is the factorization method based on the matrix or tensor decomposition [2]. As shown by [8], applying matrix factorization (MF) can lead to improved prediction results in PSP compared to traditional PSP methods. MF predicts student performance by extracting similarities among learners and items from the learner-item performance data in form of latent factors. MF creates two matrices with latent factors for each of learners and items, so there is no need to explicitly encode Q-matrix or other parameters such as Slip and Guess [14]. In addition, MF is very effective in dealing with insufficient data as it effectively captures and uses the similarities among learners and items [14]. The main limitation of MF is its lack of temporal effect as MF discards any temporal information and learns the typical performance of students at one time. Tensor Factorization overcomes this limitation; however, the running time of tensor factorization is significantly longer than MF [13], so in practice it is not used as frequently.

In this paper, we introduce a new approach called MBKT that combines BKT and MF for the task of PSP. Traditional models of combining where the predictions results of individual algorithms are stacked, would require MF to learn an implicit Q-matrix and latent factors incorporating Slip and Guess from Scratch. To fully exploit the advantages of combining BKT and MF, MBKT first utilizes BKT to capture the temporal effects of the student model using an explicit Q-matrix and parameters referring to Slip and Guess. This information is then passed on to MF, which enables the latent factor of MF to be tuned for capturing the similarities between students and items.

Our results on four benchmark datasets obtained from the DataShop platform [9] indicate that using MBKT for combining various variations of BKT and MF for PSP outperforms the base models. We also show that MBKT outperforms traditional methods of combining the results of BKT and MF using linear regression, logistic regression and Neural Networks.

2. RELATED WORK

The problem of combining different algorithms for improvement in PSP has been well studied, with contradicting results. To evaluate the effect of ensemble techniques in Intelligent Tutoring System (ITS), Baker et al. [4] selected nine different PSP individual algorithms and combined them using logistic and linear regression on a genetic dataset. Their experimental results showed that the accuracy of ensembling is mixed and slightly different from the individual algorithms. They argued that there may be three explanations for this lack of improvement: (1) use of only simple models of ensembling like linear and logistic regression, (2) use of small datasets with a limited number of learner interactions, and use of similar ensemble techniques on learning models with slight differences. Pardos et al. [10] reported that ensembling on large enough datasets will lead to promising improvements even with similar base models. However, in practice, student models rely on small datasets for training, so the results of ensemble techniques on large datasets cannot be applied directly to ITS. More recently, [12] used a knowledge graph representation to identify feasible activity scopes, which were combined to predict student performance on a learning objective in an ensemble.

Despite development of various ensembling algorithms on PSP, to the best of our knowledge, collaborative filtering algorithms have not been used in conjunction with knowledge tracing algorithms in the previous studies. Given that these two complement each other on many fronts, we attempt to extend the work of previous studies by primarily investigating the impact of combining MF as a leading collaborative filtering algorithms with knowledge tracing for PSP.

3. COMBINING KNOWLEDGE TRACING AND MATRIX FACTORIZATION

As mentioned in the previous sections, the characteristics of BKT and MF complement each other quite well. BKT utilizes the temporal behavior of learners to model their learning, while MF does so by capturing the similarities among learners and items. In addition, BKT uses an explicit Q-matrix to find the parameters related to learners including their initial knowledge of skills, the mastery probability of skills and Slip and Guess parameters. In contrast, MF uses latent factors to implicitly learn a Q-matrix and the mentioned learner-related parameters. In this paper, we propose a new model called MBKT for combining BKT [3] and MF [14] for PSP that takes advantage of how these models complement each other. We also considered two other variations of BKT as described in [6]. The first variation, BKT-CGS (Contextual Guess and Slip) model, is a variation in which Guess and Slip properties are no longer learned per skill but rather averaged across all skills and actions. The second variation, BKT-PPS (Prior Per Student) assumes a personalized prior knowledge per student. In our experiments, we used a simplified version of this model that divides students to high-performance and low-performance groups as proposed by [6]. Using MBKT, the predicted performance of leaner u on item i is predicted as follows:

In the first step, the BKT model is utilized to predict student performance using the following formula

$$O^{BKT}_{N\times M} = BKT(train\_set),$$

where o^{BKT}_{ui} presents the computed probability of the BKT model on user u answering item i correctly based on the last opportunity of u on the topic related to i.

In the second step, the error of BKT predictions for the learner-item performance is computed as follows

$$E^{BKT}_{N\times M} = O^{train}_{N\times M} - O^{BKT}_{N\times M},$$

where o^{train}_{ui} is 1 if user u has answered question i correctly in their final attempt, 0 if user u has answered question i incorrectly and Null otherwise. e^{BKT}_{ui} is the computed error of the BKT model for user u on question i.

In the third step, the error of BKT predictions for the learner-item performance is passed on to MF as input to predict the BKT prediction error using the following formula

$$O^{MF}_{N\times M} = MF(E^{BKT}_{N\times M}),$$

where o^{MF}_{ui} presents the approximated error of the BKT model on the final opportunity of user u on answering question i.

Finally, the outcome of MBKT is computed by summing the BKT predictions and predicted error of MF for BKT using

$$O^{MBKT}_{N\times M} = O^{BKT}_{N\times M} + O^{MF}_{N\times M},$$

where o^{MBKT}_{ui} represents the predicted performance of user u on question i, which is computed by MBKT.

Discussion. Using the traditional models of combining where the prediction results of individual algorithms are stacked, MF needs to learn the Q-matrix and latent factors from scratch using a random initialization. This makes the combination unlikely to fully exploit the advantage of combining BKT as a temporal model and MF as a model to draw out the similarities among learners and items. In MBKT, instead of directly stacking the prediction results of BKT and MF, BKT is utilized as the underlying algorithm to predict student performance. Then the prediction error of BKT is passed to MF as input to learn the BKT error. Insinuating the outcome of the BKT model in the input of MF enables MBKT to benefit from BKT's explicit parameterization of the learners and items including the initial knowledge, the mastery probability of skills and Slip and Guess concepts. This, in turn, would enable the latent factors of MF to further focus on modeling similarities among learners instead of trying to incorporate those parameters.

4. EXPERIMENTS

In this paper, we have discussed the benefits of combining knowledge tracing and collaborative filtering algorithms for PSP using MBKT. In this section, we aim to investigate whether use of MBKT leads to improved PSP. Our evaluation has been guided by the following two research questions.

- RQ1: Does MBKT improve the performance of PSP compared to the base models?
- RQ2: Does MBKT improve the performance of PSP compared to traditional techniques of stacking the results of BKT and MF?

For the experiments, we utilize LearnSphere [9] to find the parameters of each BKT variation using 10 fold cross-validation with Baum-Welch solver. To find the latent factors related to each MF variation, we use MyMediaLite library [5] with again 10 fold cross-validation.

4.1 Dataset

We use four data sets that are commonly used for PSP from DataShop [9] in our evaluation. The total number of interactions and students of each dataset is described in table 1.

Table 1: DataSets

Data Set	#transactions	#students
Geometry Area	6,778	59
Intelligent Writing Tutor	6,625	120
Writing 1	12,568	31
Writing 2	11,347	54

These are the results of learners' interactions with the tutoring system. As learners engage in the system, all interactions such as their success or failure, time spent on each step, etc are recorded. In these experiments, the unique interaction between learners and system is the step, which belongs to the hierarchy of *unit*, *section* and *problem*. KC defines different knowledge components for each step in the hierarchy and *Opportunity* determines the total number of times that a leaner has had on the KC related to the step. In these datasets $FirstAttempt$ is considered as the outcome of the interaction: *correct* means success and *incorrect* and *hint* show failure in that interaction.

4.2 Methods and Evaluation Metric

In our experiments, standard BKT (BKT) [3], Contextualized Guess and Slip BKT (BKT-CGS) [10], Prior Per Student BKT (BKT-PPS) [10], Standard Matrix Factorization (MF) and Biased Matrix Factorization (BMF) as described in [14] are used as the base methods.

The BKT and MF variations are combined using logistic regression (LogReg), linear regression (LinReg), Neural Networks (NN) and MBKT.

Evaluation Metric. As commonly used in evaluating the PSP algorithms, Root Mean Squared Error (RMSE) is utilized to measure the error as follows:

$$RMSE = \sqrt{\frac{1}{|D|^{\text{test}}} \sum_{(u,i)\in D^{\text{test}}} (o_{ui}^{\text{test}} - o_{ui}^{\text{prediced}})^2},$$

where o_{ui}^{prediced} is the predicted probability, o_{ui}^{test} is the real output of the instance and D^{test} is the total number of instances.

4.3 Results

Table 2 compares the RMSE of the model fit statistics related to each model for the task of PSP. In this table, Geo, IntW, HW1, and HW2 refer to Geometry Area, Intelligent Writing, Hand Writing 1, and Hand Writing 2 datasets respectively. Based on the experimental results for all datasets, there is no superiority among different BKT variations. Among the two MF variations, BMF significantly outperforms MF both as an individual algorithm and in combination with the BKT variations. For instance, the average RMSE for BMF and MF as an individual algorithm on all datasets is 0.412 and 0.427 respectively. A similar difference is observed in the combinational models. So, for the rest of discussions, we only concentrate on BMF as the collaborative filtering algorithm.

Table 2: RMSE of different learning models

Methods		**Geo**	**IntW**	**HW1**	**HW2**
BKT		0.422	0.438	0.431	0.408
BKTPPS		0.421	0.422	0.412	0.392
BKTCGS		0.419	0.438	0.431	0.407
MF		0.427	0.453	0.433	0.396
BMF		0.418	0.433	0.407	0.390
BKT -MF	LogReg	0.419	0.447	0.440	0.397
	LinReg	0.424	0.447	0.450	0.397
	NN	0.420	0.449	0.451	0.4
	MBKT	0.428	0.44	0.432	0.395
BKT -BMF	LogReg	0.417	0.421	0.406	0.391
	LinReg	0.415	0.422	0.406	0.390
	NN	0.420	0.421	0.406	0.391
	MBKT	0.411	0.418	**0.404**	0.387
BKTPPS -MF	LogReg	0.419	0.431	0.428	0.395
	LinReg	0.424	0.427	0.433	0.395
	NN	0.420	0.435	0.438	0.396
	MBKT	0.424	0.423	0.417	0.391
BKTPPS -BMF	LogReg	0.417	0.412	0.406	0.388
	LinReg	0.416	**0.411**	0.407	0.390
	NN	0.420	0.412	0.407	0.387
	MBKT	0.415	**0.411**	0.406	**0.386**
BKTCGS -MF	LogReg	0.420	0.447	0.44	0.397
	LinReg	0.430	0.447	0.405	0.397
	NN	0.421	0.449	0.452	0.4
	MBKT	0.422	0.435	0.431	0.394
BKTCGS -BMF	LogReg	0.416	0.421	0.406	0.391
	LinReg	0.415	0.422	0.406	0.399
	NN	0.421	0.421	0.406	0.391
	MBKT	**0.408**	0.418	0.405	0.387

RQ1. The results of cross-validated RMSE on all datasets indicates that for all combinations of BKT variations and BMF, MBKT achieves the best RMSE. As presented in Table 2, MBKT outperforms its base models by ≈ 10%. To evaluate the statistical significance of the improvements in predictions, Ttest is used. For each dataset, we applied Ttest on the RMSE of the best individual model and the best combination of BKT and MF using MBKT. For all four datasets, the difference between the results of the individual algorithms and MBKT was statistically significant with the computed p values smaller than 0.01.

RQ2. To answer this research question, we used the traditional stacking techniques including linear regression, logistic regression, and Neural Network to combine each of the BKT variations with BMF. Our experimental results on all datasets indicate that for each combination of BKT variations and BMF using MBKT and other stacking techniques, MBKT always outperforms the traditional stacking techniques, except for IntW where linear regression achieves the same RMSE as MBKT when combining BKTPPS and BMF. To evaluate the statistical significance of the models, we limited our comparisons to the combinations with the same base models. Our results on the four datasets indicate that with BKTPPS and BMF as the base models, MBKT and linear regression were not significantly different from one another for both Geometry Area and Intelligent Writing Tutor datasets. For the renaming 10 combinations, MBKT improve PSP with statistical significance ($p < 0.01$) compared to traditional stacking techniques.

In addition, MBKT always outperforms its base models and achieves $\approx 10\%$ improvement in the predictive power compared to its underlying BKT model. This is a significant improvement for a predicting model. In contrast, applying the traditional combining models do not always improve the predictions over those of the base models. For example, for Hand Writing 2, using logistic regression or Neural Network for combining BKT or BKT-CGS with BMF leads to poorer RMSE than BMF itself. This lack of success for traditional combining models reflects the same result is presented by [4].

5. CONCLUSION AND FUTURE WORK

In this paper, we investigated the effect of combing time-aware knowledge tracing algorithms with matrix factorization as a time-invariant collaborative filtering algorithm for PSP. Variations of Bayesian Knowledge Tracing (BKT) and Matrix Factorization (MF) were used for this task. These models complement each other's strengths and limitations quite effectively. BKT captures temporal changes in learners' behavior using an explicit Q-matrix and BKT parameters such as Slip and Guess. In contrast, MF captures similarities among learners using latent variables that implicitly encode a Q-matrix as well as learners' initial knowledge, skill mastery probability, Slip and Guess Parameters. We introduced an algorithm for combining MF and BKT, where instead of directly combining the prediction result of each individual algorithm, it first utilizes BKT as the underlying algorithm to predict student performance. It then passes the error, true values - predicted values, from BKT predictions as input to MF. Incorporating the outcome of the BKT model in the input of MF enables it to benefit from BKT's explicit parameterization including Slip and Guess concepts. This, in turn, would enable the latent factors of MF to further focus on modeling similarities among learners instead of trying to incorporate Slip and Guess parameters.

Our results on four benchmark datasets from the Datashop platform indicates that using MBKT for combining variations of BKT and MF leads to as much as 10% improvement over the base models for PSP on unseen datasets. In addition, MBKT generally provides statistically significant improvements over traditional techniques such as linear regression, logistic regression and Neural Networks for combining BKT and MF again, for PSP on unseen dataset.

There are several interesting directions to pursue in future work. Primarily, we are working on integrating our approach into an open-source, student facing adaptive learning environment called Recommendation in Personalized Peer Learning Environments (RiPPLE) [7]. Our goal is to use the proposed algorithm for predicting student performance, which in turn, is used for recommending personalized questions based on learners' current knowledge gaps.

6. REFERENCES

[1] Tiffany Barnes. The q-matrix method: Mining student response data for knowledge. In *American Association for Artificial Intelligence 2005 Educational Data Mining Workshop*, pages 1–8, 2005.

[2] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[3] 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.

[4] Ryan SJ d Baker, Zachary A Pardos, Sujith M Gowda, Bahador B Nooraei, and Neil T Heffernan. Ensembling predictions of student knowledge within intelligent tutoring systems. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 13–24. Springer, 2011.

[5] Zeno Gantner, Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. MyMediaLite: A free recommender system library. In *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys 2011)*, 2011.

[6] SM Gowda, RSJD Baker, Z Pardos, and NT Heffernan. The sum is greater than the parts: ensembling student knowledge models in assistments.

[7] Hassan Khosravi. Recommendation in personalised peer-learning environments. *arXiv preprint arXiv:1712.03077*, 2017.

[8] Hassan Khosravi, Kendra Cooper, and Kirsty Kitto. Riple: Recommendation in peer-learning environments based on knowledge gaps and interests. *JEDM-Journal of Educational Data Mining*, 9(1):42–67, 2017.

[9] Kenneth R Koedinger, Kyle Cunningham, Alida Skogsholm, Brett Leber, and John Stamper. A data repository for the edm community: The pslc datashop. *Handbook of educational data mining*, 43, 2010.

[10] Zachary A Pardos, Sujith M Gowda, Ryan SJd Baker, and Neil T Heffernan. The sum is greater than the parts: ensembling models of student knowledge in educational software. *ACM SIGKDD explorations newsletter*, 13(2):37–44, 2012.

[11] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 811–820. ACM, 2010.

[12] Martin Stapel, Zhilin Zheng, and Niels Pinkwart. An ensemble method to predict student performance in an online math learning environment. In *EDM*, pages 231–238, 2016.

[13] Nguyen Thai-Nghe, Lucas Drumond, and Tomás Horváth. Matrix and tensor factorization for predicting student performance.

[14] Nguyen Thai-Nghe, Lucas Drumond, Tomáš Horváth, Artus Krohn-Grimberghe, Alexandros Nanopoulos, and Lars Schmidt-Thieme. Factorization techniques for predicting student performance. *Educational recommender systems and technologies: Practices and challenges*, pages 129–153, 2011.

Defining Personalized Writing Burst Measures of Translation Using Keystroke Logs

Mo Zhang, Jiangang Hao, Paul Deane, and Chen Li
Educational Testing Service
660 Rosedale Road
Princeton, New Jersey, 08540
{mzhang, jhao, pdeane, cli}@ets.org

ABSTRACT

Looking only at the final writing does not tell us how the students pursued the writing task. The keystroke logs can help us disambiguate what we have observed from the final essays. In this study, we focus on the analysis of writing bursts, which are defined as sequences of rapid text production without long pauses. Compared to the existing measures, the new personalized burst measures showed stronger association with essay quality, a weaker correlation with keyboarding skill, and moderately strong cross-task reliability.

Keywords

Response process, Keystroke log, Writing burst, Translation process, Hierarchical clustering

1. INTRODUCTION

There is a growing number of recent educational research studies that use timing and process information for assessment purposes [6]. Response process data provide information about students' proficiency and performance that is not accessible by examining the final answers alone. As elucidated in the *Standards* [1], the processes that test-takers undertake in responding to an item or an assessment provide validity evidence for evaluating the meangingfulness of the inferences made on test scores. In this paper, we focus on a specific type of response process data that is collected via logging the keystrokes and mouse movements during a writing task. This work is part of a larger research and development project at Educational Testing Service (ETS) that aims to understand, analyze, validate, and ultimately report writing processes in educational and assessment contexts.

Generally speaking, analyzing writing process data starts with the development of a set of feature variables that represent the writing process quantitatively. Among various feature variables that have been used to characterize the writing process, sequences of fast text production, or bursts, have been identified as providing important information about the fluency and efficiency of individual writing processes [2]. Although there is a consensus about the importance of bursts, there are many ways to define a burst in practice, depending on the location where bursts are allowed to end (between characters, or between words), and on the length of a pause that is considered long enough to count as a burst boundary. Here, we introduce a specific approach to defining bursts at the word level, **personalized bursts**, which can be obtained by identifying the optimal number of clusters for each individual, by applying a hierarchical clustering analysis to individual inter-word intervals. The results of this study highlight the potential that keystroke logs of writing have for educational assessment applications.

1.1 Keystroke Logging of Writing Process

When writing tasks are delivered on a computer, we can record the processes by which a student produces his/her essay response through keystroke logging. In this study, we used a keystroke logging system developed at ETS – one primarily intended for large-scale digital administrations to support classroom instruction and educational assessment [5]. A keystroke logging system records all changes to the text buffer while a student is writing, along with associated time stamps. The gap-time between keystrokes, which is often termed "silence" in speech, is usually called a "pause" in the analysis of keystroke logs. The entire text production process can be precisely reconstructed from the keystroke log. Some examples of the key information tracked by the ETS system [5] are: type of action (e.g., insert, delete), length of action (e.g., with regard to the number of time elapsed), location of action (e.g., between words), and timepoint of action (e.g., at the start of a writing session).

1.2 Cognitive Model of Writing Process

The current study is guided by the cognitive framework proposed by Hayes [7]. In this framework, the writing process is represented as a multidimensional construct including four connected dimensions. Each dimension has its distinct behavioral and temporal features. For example, idea generation and task preparation (i.e., **proposing**) are generally associated with pauses at the start of writing and at sentence boundaries where writers stop and think about what to say. Fluency of putting ideas into language (i.e., **translating**) relates to the length of bursts. Orthographic proficiency and motor skill (i.e., **transcribing**) typically relates to pauses inside a word and to edits designed to make immediate corrections to typos. And, when writers are editing and reviewing (**evaluating**), they are more likely to jump to different locations in the text to make changes or replace chunks of existing text with new content. With keystroke logs, these theoretically-defined subconstructs of the writing process can be estimated separately (see, for example, [9]).

2. RESEARCH PROBLEM AND QUESTION

In this study, we analyzed one of the most important features in the psycholinguistic literature – bursts of rapid text productions. Bursts are defined as stretches of rapid, consecutive text production without major interruptions, where an interruption point is signaled by a long pause. The literature has suggested that various burst measures, including burst size, burst frequency and maximum burst length, are indicative of a writer's text production fluency. One difficulty in using existing burst measures to make inferences about writing fluency, however, is that the properties of bursts can be affected by more than one cognitive process, including *both* the ability to translate ideas into language (translating) *and* the ability to put words on paper (transcribing). In practice, the transcription subprocess can be approximated by typing speed. Ideally, we would like a measure of fluency, separate from typing speed, that reflects the speed with which a writer can generate ideas and put them into words.

Previous research recommends a 2/3 second pause for inter-key intervals and/or a 2 second pause for inter-word intervals, to indicate a burst boundary [2]. This approach applies a single pre-determined fixed threshold for every text produced by every writer. It is simple and straightforward; yet one disadvantage of this approach is that the resulting burst measures do not control for keyboarding skill. A slow typist may seem disfluent without actually having weaker abilities to generate ideas or to put them into words.

The existing literature provides one alternative approach to defining writing bursts which is less strongly linked to typing speed [5]. In this approach, the thresholds for burst boundaries change as the composition proceeds. Burst boundary thresholds are calculated on-the-fly for inter-word intervals (IWI) using all IWI information collected so far, where the thresholds (in current practice) are defined as being four times the median across previous IWIs. This method generates personalized thresholds across each essay response, and yields relatively long bursts that are less likely to change due to differences in typing speed. Due to its time-adaptive nature, this method does not fully consider the pause patterns throughout the writing process. This approach can also be sensitive to a writer's composition strategy and, from previous analyses, generates burst measures only weakly related to the quality of an individual's writing.

Our goal in this study is to find a better way of to define the cut point between bursts, one that is more sensitive to linguistic processes (translation), but less sensitive to typing speed (transcription). The approach we have investigated is also personalized to individual writers. Unlike the above-mentioned time-adaptive thresholds, this new approach uses all inter-word pause information together, thereby fixing the threshold for each individual instead of changing it as writing proceeds. We empirically compared the performance of this new method (fixed and personalized) to the two baseline methods (fixed for all persons vs. adaptive by person). For simplicity, we will refer to the new method as "personalized," and the two existing ones as "fixed" and "adaptive." Our research question is: can we find a criterion for setting personalized burst boundaries that better characterizes an individual's writing behavior?

3. METHOD

3.1 Participants and Instrument

The participants were 1,351 6^{th}-8^{th} grade students in U.S. middle schools. The data collection procedure can be found in [8]. Three scenario-based persuasive writing assessments [3] were used, each containing one essay-length writing task. Each student took two of the three assessments. The three assessments were strictly parallel with the only difference in the scenario presented to the students. The Service Learning (SL) scenario: What would be the best choice of service learning project for a class to carry out? The Culture Fair (CF) scenario: What would be the best theme for a school culture fair? The Generous Gift (GG) scenario: What is the best way for a school to spend a large sum of money provided by a generous donor? The student sample sizes for each assessment, respectively, are 842 (SL), 831 (CF), and 557 (GG). Each essay response was graded by a human rater on two scoring rubrics. The first rubric evaluates the standard academic English writing skills, such as grammar and word usage. The second rubric evaluates the key elements required for persuasive writing. A total essay score is then calculated as the sum of the two rubric scores.

3.2 Personalized Burst

The following single-sentence example illustrates the general scheme of searching for an optimal personalized burst threshold (Figure 1). In this example, each IWI is marked. The longest IWI is .80 sec., lying between the words "think" and "that." The second longest IWI is .60 sec., lying between "should" and "cook," and so on. If the burst boundary threshold is set at .80 sec. (i.e., the longest IWI, Level 1), the writing process will be broken into two bursts: "I think" would be the first burst, and the remainder of the sentence would be the second. If we lower the threshold to Level 2, then the writing process would be broken into three bursts. In the extreme case, for this example, using a threshold of .05 sec. (Level 5) would break this short sentence into ten one-word bursts.

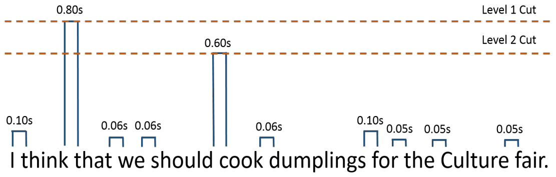


Figure 1: Impacts of Burst Cut Threshold on Translation Process: A One-Sentence Illustration

This approach to setting burst boundaries can easily be generalized to complete keystroke logs. Figure 2 shows a dendrogram based on the single-linkage of the IWIs from a full essay. It is clear that, if we make the threshold too low, the translation process will become rather fragmented. The goal here is to search for an optimal cut threshold, by which one can meaningfully break the translation process into an optimal number of bursts.

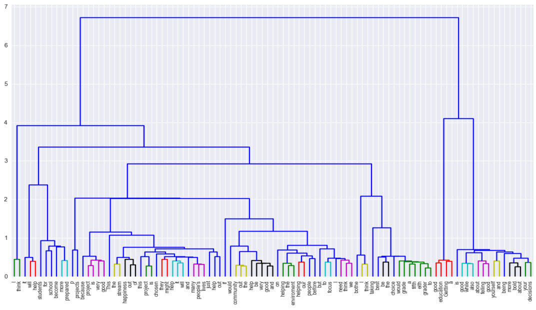


Figure 2: Dendrogram of IWI in a Complete Essay

From the perspective of hierarchical clustering analysis, this is essentially a problem dealing with the optimal number of clusters. We used the Calinski-Harabasz (CH) index, $CH(k) = \frac{B(k)}{W(k)} \frac{n-k}{k-1}$, to determine the optimal number of clusters [4], where n refers to the number of elements, k is the number of clusters, and $B(k)$ and $W(k)$ are the between- and within- cluster variances respectively corresponding to k clusters. The optimal number of clusters corresponds to the threshold value that leads to the maximum CH value.

3.3 Burst Features and Evaluation

Once we have determined how to define burst boundaries, we can calculate summary statistics for each response. The literature suggests that average and maximum burst length (BL) is of particular interest. We expect more fluent writers to produce longer bursts, and less fluent writers to produce shorter bursts. We therefore computed the average personalized BL and the maximum personalized BL for each essay. To evaluate the performance of the personalized burst features, we compared them with summary statistics for two existing burst definitions: fixed bursts (2/3 sec for IKI) and adaptive bursts (4x the median IWI). In particular, we examined their correlations with essay scores and typing speed. As discussed above, a measure of BL that separates translation fluency from transcription fluency should not be too strongly associated with keyboarding (KB) skills. For this purpose, we generated a KB measure designed to provide a fairly pure measure of typing speed. This measure was based on how quickly a writer typed extremely common English words. We followed the practice suggested in [10], in which the typing speed is calculated as the median value of characters per second across all valid words on the most common 100 English words and their inflections. Finally, since most students submitted two essay responses, we examined the cross-prompt reliability of the different burst measures.

4. RESULTS

4.1 Association with Quality and Typing

In Table 1, we examined various burst length measures' correlations with essay quality as evaluated by human raters and with typing speed. We also included two reference features: Essay Length (the number of words in the essay) and Time on Task, both of which are known to be moderately correlated with writing quality.

Two findings are worth highlighting. First, we observed that the personalized method for defining bursts yielded stronger correlations with human score on both rubrics, compared to fixed or adaptive methods. For rubric 1, the average BL and maximum BL from the new threshold correlated with rubric 1 in the magnitude of .50s, whereas the others had correlations in the magnitudes of .20s or mid .30s. The correlations were not as strong as Essay Length with rubric 1, but were greater than the total writing time. Similar observations apply to the content rubric (rubric 2), where the new burst features outperformed the existing burst features.

A second critical finding is that the average and maximum personalized BL correlated only moderately with keyboarding skill as reflected by the speed of typing common words. From Table 1, we can see that although the BL statistics derived from the adaptive burst definition showed the weakest correlation with keyboarding skills, their correlations with the writing quality was poor, particularly with the content score, only 0.11 and 0.26 in magnitude. The personalized burst features, by contrast, showed moderate correlations with keyboarding, 0.45 for mean BL and 0.48 for max BL, but displayed high correlations with essay score. This is a desirable property since we would like to separate the contributions of the translating subprocess from those of the transcribing subprocess, while still obtaining good information about overall writing quality.

4.2 Cross-Prompt Reliability

The availability of double essays submitted by students allowed for analysis of feature consistency across writing tasks. Table 2 shows the results of the cross-prompt reliability for each burst measure. Higher feature reliability is preferable, since we would like the burst features to provide stable estimates of individual traits. Of course, in reality, not all writing tasks are exactly equivalent to each other, so some variations are expected. The greatest cross-prompt reliability was observed with the burst measures that used a fixed threshold of 2/3 of a second to define burst boundaries. The resulting cross-prompt correlations were in the range of .80s and .90s, which spoke to the stability of these features. They achieved about the same level of reliability as the keyboarding measure. However, the 2/3 second threshold is rather low, and yields very short bursts. It is possible that this low, fixed threshold essentially cuts off the long tails where most of the pauses related to planning and idea generation take place, reducing the fixed burst measures to measures of keyboarding fluency. By contrast, the personalized burst measures showed moderately strong cross-prompt consistency. While the resulting measures are less reliable than the fixed burst measures, they were more reliable than the adaptive burst measures, and had generally higher correlations with essay length, total writing time, and essay total score.

5. DISCUSSION

The ultimate goal of our research is to identify features that more cleanly separates out different skills – translation, transcription, idea generation, and evaluation – and allow us to estimate individual parameters for each of these processes. In this paper, we introduced a new personalized burst based on a threshold optimized against the overall clustering pattern of the IWIs, with the goal of providing more direct measurement of translation fluency, and less direct measurement of transcription fluency. We compared the performance of the burst features obtained from different burst definitions

Table 1: Pearson Correlations of Burst Length Measures with Scores and Typing (SL Form)

		Basic	Content	Essay Total	Keyboarding
Threshold	Feature	(n=540)	(n=529)	(n=529)	(n = 528)
Personalized	Avg. BL (word)	.50	.32	.46	.45
	Max BL (word)	.51	.33	.47	.48
Fixed (2/3 sec.)	Avg. BL (word)	.25	.14	.22	.61
	Med. Log of BL (char.)	.34	.21	.31	.75
Adaptive (4x median)	Avg. BL (word)	.21	.11	.18	-.01†
	Max BL (word)	.37	.26	.36	.22
	Essay Length (word)	.66	.47	.63	.30
	Time on Task (sec.)	.40	.39	.45	-.13

Note: All but one correlations are statistically significant at $p < .0001$ level. †: significant at $p < .05$ level. Results reproted in this table are from the SL assessment form. BL = Burst Length

Table 2: Cross-Prompt Reliability of Burst Length Measures

		CF–GG	CF–SL	GG–SL
Threshold	Feature	(n=437)	(n=216)	(n=203)
Personalized	Avg. BL (word)	.66	.55	.69
	Max BL (word)	.52	.48	.55
Fixed (2/3 sec.)	Avg. BL (word)	.89	.87	.81
	Med. Log of BL (char.)	.91	.93	.87
Adaptive (4x median)	Avg. BL (word)	.59	.63	.51
	Max BL (word)	.45	.48	.46
	Essay Length (word)	.63	.62	.66
	Time on Task (sec.)	.44	.45	.41
	Essay Score (2-10)	.48	.53	.55
	Keyboarding (char/sec)	.89	.91	.90

Note: All Pearson correlations are statistically significant at $p <$ *.0001 level.*

and showed that the personalized burst feature was only moderately correlated with keyboarding skills, while being more strongly correlated with essay scores than other methods for defining burst thresholds. The personalized burst feature also showed reasonable cross-prompt correlations. All of these characteristics indicate that the personalized burst feature is a promising candidate to replace existing burst definitions in future writing process studies.

However, we emphasize that the personalized burst feature introduced in this paper is capable of further optimization. For example, we used the CH index to determine the optimal threshold, which, based on our experimentation, tended to produce short bursts of only a few words each. Use of other criteria might lead to slightly larger burst sizes without damaging performance. It is also worth noting that our conclusions may be limited by the nature of the studied sample. We do not know how these features shift in their performance by age, and have only begun to explore how they shift between demographic groups. Presumably, in a population that was more fluent in keyboarding, it might be easier to use bursts as a pure measure of fluency, while in an elementary school population, we might find it much more difficult to separate translation from transcription. Overall, our results suggest that there is considerable room to create improved methods for measuring student writing processes using modern statistical and data mining methodologies.

6. ACKNOWLEDGMENTS

We thank our colleagues Peter van Rijn, Dan McCaffrey and Randy Bennett for providing technical advice on this study.

7. REFERENCES

[1] AERA, APA, and NCME. *Standards for educational and psychological testing.* American Educational Research Association, Washington D. C., 1999.

[2] R. A. Alves, S. L. Castro, L. de Sousa, and S. Strom̆qvist. Influence of typing skill on pauseâĂŞexecution cycles in written composition. In M. Torrance, L. van Waes, and D. Galbraith, editors, *Writing and Cognition: Research and Applications*, pages 55–65. Elsevier, Amsterdam, 2007.

[3] R. E. Bennett, P. Deane, and P. W. van Rijn. From cognitive domain theory to assessment practice. *Educational Psychologist*, 51:82–107, 2016.

[4] T. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3:1–27, 1974.

[5] P. Deane, G. Feng, M. Zhang, J. Hao, Y. Bergner, M. Flor, M. Wagner, and N. Lederer. Generating scores and feedback for writing assessment and instruction using electronic process logs. *US Patent and Trademark Office. Application No. 14/937,164*, 2016.

[6] K. Ercikan and J. W. Pellegrino. *Validation of Score Meaning for the Next Generation of Assessments: The Use of Response Processes.* Taylor & Francis, 2017.

[7] J. R. Hayes. Modeling and remodeling writing. *Written communication*, pages 369–388, 2012.

[8] P. van Rijn, J. Chen, and Y. Yan-Koo. Statistical results from the 2013 cbal english language arts multistate study: Parallel forms for policy recommendation writing. ETS Research Memorandum RM-16-01, ETS, Princeton, NJ, 2015.

[9] M. Zhang and P. Deane. Process features in writing: internal structure and incremental value over product features. ETS Research Rep. RR-15-27, ETS, Princeton, NJ, 2015.

[10] M. Zhang, G. Feng, and P. Deane. Investigating an approach to evaluating keyboarding fluency. To be submitted, ETS, Princeton, NJ.

Diverse Learners, Diverse Motivations: Exploring the Sentiment of Learning Objectives

Nigel Bosch
University of Illinois at Urbana-Champaign
pnb@illinois.edu

R. Wes Crues
University of Illinois at Urbana-Champaign
crues2@illinois.edu

Najmuddin Shaik
University of Illinois at Urbana-Champaign
shaik@illinois.edu

ABSTRACT

Massive Open Online Courses (MOOCs) have become increasingly popular in recent years, enabling millions of students worldwide to pursue their educational objectives in new ways. However, little is known about the nature of the reasons why students enroll in courses, and how those reasons differ across demographic groups. In this paper we explore the connection between student engagement and the sentiment of their self-reported reasons for enrolling in MOOCs. We found that there were significant differences in sentiment between demographic groups, and that sentiment of enrollment reasons had small—but consistent—power to predict future course engagement level (Spearman's $\rho = .102$). Finally, we discuss the implications of these findings for future student modeling research in MOOC contexts, particularly for students with different backgrounds.

Categories and Subject Descriptors

H.4 [**Computers and Education**]: Computer Uses in Education

Keywords

Sentiment, MOOC enrollment, demographic differences

1. INTRODUCTION

Recent years have seen an explosion in the availability of massive open online courses (MOOCs), and millions of new students have enrolled in them [8]. MOOCs offer worldwide access to high-quality courses offered by prestigious institutions, and as such they attract a diverse group of learners from around the globe. These students are often even more diverse than typical university student bodies, in part because of the unique affordances MOOCs provide: low cost and flexible schedules.

However, reasons students enroll are complex and especially difficult to define in MOOC-style learning contexts. Furthermore, enrollment reasons may systematically differ between different courses and student demographics [5]. In this poster, **we focus specifically on the sentiment aspect of students enrollment reasons**. Particularly, we explore whether sentiment of reasons for enrolling differs across course and demographic dimensions, and question how sentiment of enrollment reasons relates to students' levels of engagement in a course. Sentiment in this context refers to aspects such as positivity, fear, certainty, and others that can be inferred from text. One might also expect, for example, that the amount of positivity inherent in students' stated reasons for enrolling relates to how long they persevere in a course.

Previous work has also found that demographics relate to MOOC outcomes (e.g., gender relates to persistence in MOOCs [4]). Given that there were 78 million students in almost 10,000 MOOCs [8] in the year 2017, it is tremendously important to understand all potential indicators of student engagement and student needs. This includes the emotional cues contained in the goals they may express when enrolling in a course.

Wladis et al. analyzed approximately 27,800 students who enrolled in online STEM (science, technology, engineering, and math) classes [11]. They found key differences in enrollment rates, concluding that non-traditional students were significantly more likely to enroll in online courses than their peers. Furthermore, they found that female students performed less well in online environments than in face-to-face learning, but older students enjoyed greater success online [12].

Robinson et al. [7] asked students enrolling in a MOOC to "provide one or two specific examples of how you think what you will learn in this class will apply to your life." They extracted frequent word unigrams and bigrams, and trained a logistic regression classifier to predict whether students would drop out of the MOOC or not. Their model was statistically better than chance as measured by area under the receiver operating characteristic curve (AUC)—specifically, AUC = .564 (versus .500 chance level). Additionally, they found that including student demographics as predictors improved model accuracy to AUC = .598. This study demonstrated that linguistic aspects of the reasons students state for enrolling in a class can modestly predict course outcomes, but that student demographics are also worth considering.

In this study we apply non-parametric statistics and machine learning methods to explore the relationships between the sentiment of students' reasons for enrolling in MOOCs, student demographics, and course engagement. We cross-validate analysis across courses and demographic variables to answer three research questions: 1) How does sentiment of enrollment reasons differ across student demographics? 2) Does sentiment of enrollment reasons predict the level of course engagement? and 3) Is sentiment of enrollment reasons equally predictive of engagement across different courses and demographics?

2. METHOD

We analyzed data from five different MOOCs offered on the Coursera platform[1]. These included *Creative, Serious, and Playful Science of Android Apps*, *Introductory Organic Chemistry*, *Subsistence Marketplaces*, *Introduction to Sustainability*, and *E-Learning Ecologies*. We queried students for demographic information, including age range and gender[2], and asked them to provide their reasons for enrolling in the course by writing an answer to the open-ended prompt "Why are you taking this course? What do you hope to get out of it?" Of 37,178 students who enrolled and responded to at least one question, 9,327 responded in English to all questions.

2.1 Sentiment of Enrollment Reasons

We extracted sentiment from students' written reasons for enrolling in MOOCs with the SEANCE (SEntiment ANalysis and Cognition Engine) tool [3]. SEANCE provides indices of sentiment derived from a collection of eight different databases of words, where each word is associated with a sentiment. SEANCE also provides 20 *component scores*, which are derived from principal components analysis and have interpretable labels based on the indices the components are derived from. These component scores compose the sentiment-based feature space in which we represent students' reasons for enrolling. Given the large ratio of students to features (approximately 450:1), we did not perform feature selection.

Sentiment components provided by SEANCE were not normally distributed. Thus, for comparisons involving sentiment we calculated non-parametric statistics. To compare sentiment across genders, we coded gender as a number and computed Spearman's rho (ρ) correlations between gender and sentiment components. This analysis permits testing for significant differences between genders as well as providing an estimate of the effect size (ρ ranging from -1 to 1). Age groups are categorical, but strictly ordered, so ρ is an appropriate measure for the relationships between age groups and sentiment components as well. Geographical areas are not strictly orderable in a meaningful way, so we could not measure ρ across all geographical areas together.

2.2 Prediction of Engagement from Sentiment

In this study we adopt a multi-level engagement definition to distinguish students who are only active during a few weeks of the course (≤ 2 weeks), versus those who engage with the course for some time but not the entire set of content ($3-5$ weeks), and those who complete essentially all of the course ($6-8$ weeks).

We predicted engagement from sentiment components by training a random forest [2] machine learning model using *scikit-learn* [6]. Random forests work by training a large number of small tree models (i.e., a forest) on random subsamples of data. Random forest models make no assumptions about the distribution of the data, as a Gaussian model does, for instance. This is a key consideration given the non-normal distributions of sentiment components. Our definition of engagement is also multi-level, and is thus a multiclass problem for which random forests are suited.

Predictive student models are frequently evaluated with accuracy metrics suited for binary classification problems (e.g., Cohen's κ, F_1). However, in this study the prediction target (engagement) has three strictly-ordered levels. Therefore, we evaluate model accuracy with Spearman's ρ.

We utilized different cross-validation approaches to answer the research questions in this paper. In each approach, we split data into training and testing data, trained a random forest model (optimized on training data only), and evaluated the model by its ability to predict the unseen the testing data. We repeated this process iteratively until every student (data point) had been in the testing data exactly once.

3. RESULTS

In this section we present results for our three research questions, with explanatory methods for the first research question and predictive models for the second and third questions.

RQ1: How does sentiment of enrollment reasons differ across student demographics? The number of students analyzed was large (9,327). Thus, many correlations between sentiment components and demographic variables were highly statistically significant (25 of 60 correlations with $p < .001$) even after Benjamini-Hochberg corrections for multiple tests [1]. Therefore, we report only the largest five correlations for the sake of conciseness (Table 1).

In general, females expressed more sentiment in their stated reasons for enrolling. In fact, mean $\rho = .047$ across all 20 sentiment components. The largest difference between genders was in the SEANCE economy component, which consists of words from manually-curated lists of nouns and adjectives related to economical concerns [9]. Females were coded as 1, so the positive correlation ($\rho = .106$) indicates that females expressed more economy-focused words than males.

Both female students and older students expressed more fear and disgust in their reasons for enrolling ($\rho = .104$ and .101 respectively). Older students also appeared to express more sentiment than younger students, based on the largest five correlations in Table 1. However, mean correlation across all 20 sentiment components for age groups was just $\rho = .007$, indicating that the larger sentiment differences in Table 1 were offset by many smaller negative correlations (12 of 20 correlations were negative).

RQ2: Does sentiment of enrollment reasons predict the level of course engagement? We trained predictive models with four-fold cross-validation to answer this research question. Predictions were significantly better than chance ($\rho = .102$, $p < .001$), confirming the hypothesis of the research question. Additionally, accuracy was consistent across folds, ranging from $\rho = .093$ to $\rho = .116$. This serves as a baseline for research question 3, which explores prediction variance across demographics and courses to quantify generalization.

[1] https://www.coursera.org

[2] Gender responses included female, male, and other, but after filtering the dataset (as described in Section 2) the only responses were female and male.

Table 1: **Differences between enrollment reason sentiment components for students with different demographics.**

Sentiment component	Spearman's ρ
Gender (female = 1)	
Economy	.106
Fear and disgust	.104
Joy	.085
Politeness	.082
Virtue adverbs	.081
Age group	
Fear and disgust	.101
Respect	.066
Certainty	-.057
Politeness	.056
Objects	.052

Overall accuracy was modest. It is, however, notable that the prediction was better than chance, given the difficulty of the problem—predicting student engagement before the course even begins. In comparison, Robinson et al. [7] trained models to predict course dropout from extensive text features. They achieved a similar degree of accuracy (AUC = .564 versus .500 chance level), despite using features capturing all types of words and word pairs—not just sentiment words.

RQ3: Is sentiment of enrollment reasons equally predictive of engagement across different courses and demographics? We re-trained the classification model in research question 2 to measure generalization by cross-validating across courses and demographics instead of four-fold cross-validation. Table 2 details the results.

Course-level cross-validation resulted in notably lower accuracy than four-fold cross-validation (overall ρ = .066 versus .102), indicating that sentiment of students' enrollment reasons was less predictive across courses. Accuracy, when testing on the *Android Apps* course, was particularly notable, in that it was not significantly above chance despite having 3,050 students. Conversely, engagement prediction did generalize well from other courses to the *Subsistence Marketplaces* course (ρ = .119).

Models did not generalize well across genders compared to the four-fold model that ignored gender (overall ρ = .073 versus .102). However, female and male results were similar (ρ = .085 and .067 respectively).

Conversely, predictive models generalized well across age groups (overall ρ = .103). Accuracy was consistent as well, ranging from ρ = .078 to .133. Because there was little fluctuation in ρ across age groups, it follows that age group and sentiment were unrelated, at least with respect to engagement (though there were differences in sentiment overall; see Table 1).

There was a large degree of variation in prediction accuracy across different geographical regions, ranging from ρ = -.019 to .368. However, several of these regions were represented by only a few students (as low as 12), so results should be approached with an appropriate degree of caution. Overall accuracy was notably lower than the four-fold model (ρ = .070 versus .102), indicating that sentiment of enrollment reasons was unequally predictive across regions.

Table 2: Classification accuracy (Spearman's ρ) when predicting course engagement generalizing across courses and demographics.

Cross-validation approach	ρ	p-value	N
Leave one course out			
Android Apps	-.009	.637	3,050
E-Learning Ecologies	.091	.009	830
Organic Chemistry	.021	.565	782
Subsistence Marketplaces	.119	.001	728
Sustainability	.052	.001	3,937
Overall result	*.066*	*.000*	*9,327*
Leave one gender out			
Female	.085	.000	4,061
Male	.067	.000	5,266
Overall result	*.073*	*.000*	*9,327*
Leave one age group out			
< 18	.133	.175	105
18-24	.105	.000	1,484
25-29	.078	.001	1,931
30-39	.082	.000	2,471
40-49	.112	.000	1,457
50-59	.108	.000	1,096
> 59	.119	.001	783
Overall result	*.103*	*.000*	*9,327*
Leave one region out			
Africa	.171	.177	64
Asia	.081	.089	444
Australia	-.019	.863	81
Central and South America	.124	.110	167
Europe	.101	.003	865
North America	.074	.000	7,694
Other	.368	.239	12
Overall result	*.070*	*.000*	*9,327*

4. GENERAL DISCUSSION

We expected gender, age, and geographical variation among students would relate to the sentiment they express in their reasons for enrolling. For example, there were clear differences in rates of enrollment for females and males depending on course topic, especially for the *Android Apps* course (much higher male enrollment). Such enrollment differences could be driven, in part, by sentiment at the time of enrollment. In fact, we found differences in the sentiment of students' reasons for enrolling among students of differing genders and ages, though less difference across geographical regions. Both female and older students shared an increased expression of fear and disgust compared to their male and younger student peers, respectively (Table 1).

We also expected sentiment of enrollment reasons to be predictive of course engagement, though not to a large degree since there are other possible factors at play (e.g., individual differences, unexpected life events, quality of instruction). Indeed, we found a predictive random forest classification model based on sentiment was significantly better than chance when predicting three levels of engagement. However, prediction accuracy was greatly impacted by geographical region (Table 2). It is possible that the results are indicative of regional differences between students. For example, cultural expectations could impact expression of sentiment, as could use of English as a second language—as is likely the case for many students outside North America.

Our findings suggest design choices for MOOCs with data-driven interventions to improve retention (e.g., [10]). Students' enrollment sentiment could be analyzed to predict engagement or enrollment with the goal of driving interventions. Given modest accuracy, such interventions should be "fail-soft", but would also be combined with existing models in an ensemble to target interventions more accurately.

4.1 Limitations and Future Work

In this study we explored the sentiment of students' reasons for enrolling in MOOCs. However, some students might stay in MOOCs for different reasons than why they enrolled. In other words, they might discover unexpected value in a MOOC that extends or replaces the original reasons they had for enrolling. Future work should extend this research to consider how students' reasons for remaining in a MOOC evolve over time, and in particular how sentiment of their reasons changes in response to successes and failures they experience.

Some of our results were also limited by sample size despite the large number of students considered. Certain comparisons between demographic groups and predictive model generalization across demographics would have benefited from more data. For instance, there were only 64 students from Africa (Table 2) in our data, and even though results suggest the engagement prediction model generalized well to these students, it is unclear without additional data. Future work should focus on groups underrepresented in MOOCs so that they are not "left behind" by models and analyses tuned for traditional majority students.

5. CONCLUSION

In this paper we examined the sentiment of students' self-reported reasons for enrolling in MOOCs, and found that there were demographic differences. Although those differences were small, they consistently predicted some of the variation in course achievement across five different MOOCs. Our findings will lead to future work understanding students' learning objectives, especially with respect to better understanding how learners from different backgrounds approach courses differently. It is our objective that this will eventually lead to MOOCs that are designed to support the needs of all students.

References

[1] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[2] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct. 2001. DOI: 10.1023/A:1010933404324.

[3] S. A. Crossley, K. Kyle, and D. S. McNamara. Sentiment Analysis and Social Cognition Engine (SEANCE): An automatic tool for sentiment, social cognition, and social-order analysis. *Behavior Research Methods*, 49(3):803–821, June 2017. DOI: 10.3758/s13428-016-0743-z.

[4] R. W. Crues, G. M. Henricks, M. Perry, S. Bhat, C. J. Anderson, N. Shaik, and L. Angrave. How do gender, learning goals, and forum participation predict persistence in a computer science MOOC? *ACM Transactions on Computing Education.*

[5] R. F. Kizilcec and E. Schneider. Motivation as a lens to understand online learners: Toward data-driven design with the OLEI scale. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 22(2):6:1–6:24, Mar. 2015. DOI: 10.1145/2699735.

[6] 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. *Journal of Machine Learning Research*, 12:2825–2830, Nov. 2011.

[7] C. Robinson, M. Yeomans, J. Reich, C. Hulleman, and H. Gehlbach. Forecasting student achievement in MOOCs with natural language processing. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, LAK '16, pages 383–387, New York, NY, USA. ACM, 2016. DOI: 10.1145/2883851.2883932.

[8] D. Shah. By The Numbers: MOOCS in 2017, Jan. 2018. URL: https://www.class-central.com/report/mooc-stats-2017/.

[9] P. J. Stone, D. C. Dunphy, and M. S. Smith. *The general inquirer: A computer approach to content analysis.* MIT Press, Oxford, England, 1966.

[10] J. Whitehill, J. J. Williams, G. Lopez, C. A. Coleman, and J. Reich. Beyond prediction: First steps toward automatic intervention in MOOC student stopout. In *Proceedings of the 8th International Conference on Educational Data Mining (EDM 2015)*, pages 171–178. International Educational Data Mining Society, 2015.

[11] C. Wladis, K. M. Conway, and A. C. Hachey. The online STEM classroom—Who succeeds? An exploration of the impact of ethnicity, gender, and non-traditional student characteristics in the community college context. *Community College Review*, 43(2):142–164, Apr. 2015. DOI: 10.1177/0091552115571729.

[12] C. Wladis, A. C. Hachey, and K. Conway. Which STEM majors enroll in online courses, and why should we care? The impact of ethnicity, gender, and non-traditional student characteristics. *Computers & Education*, 87:285–308, Sept. 2015. DOI: 10.1016/j.compedu.2015.06.010.

Online Quizzes Predict Final Exam Scores Better Than Hand-Graded On-Paper Quizzes

Byron Drury
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139
bdrury@mit.edu

Sunbok Lee
University of Houston
4800 Calhoun Rd
Houston, TX 77004
sunbok@mit.edu

Chandralekha Singh
University of Pittsburgh
4200 Fifth Avenue
Pittsburgh, PA 15260
singh@bondi.phyast.pitt.edu

David Pritchard
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139
dpritch@mit.edu

ABSTRACT

To investigate the reliability and validity of online exams as a replacement of traditional on-paper examinations, we compared simultaneously given online and traditional on-paper weekly quizzes in an introductory Newtonian mechanics course. The online quizzes were composed of approximately 10 questions and the on-paper quizzes of one or two problems hand-graded with a rubric. Both correlated comparably with the traditional long problems on the hand-graded final exam, but the online quizzes correlated better with both the concept questions and the Mechanics Reasoning Inventory [3] on the final exam. The overall correlation of the online quizzes with the final examination was high, r = 0.88. We conclude that online quizzes are a better overall measure of student ability in mechanics, likely due to a number of factors: research-developed online questions, broader coverage due to having more questions, absence of grading error, and an observed greater reliability (freedom from random testing error). Online examinations offer advantages such as immediate feedback for the students, reduction of grading errors, stable year to year comparisons of student knowledge, and reduction of faculty and staff time spent grading. Additionally, their adoption would provide key outcomes data that would benefit EDM studies of student knowledge and learning.

1. INTRODUCTION

A central objective of EDM is to assess student knowledge and skills, ideally in real time. Unfortunately, in most on-land campus settings, assessment is administered on paper, making it difficult to get detailed student response data into digital form for subsequent data mining. “Traditional” hand-graded, open response symbolic problems with rubric-based partial credit grading have long been considered the gold standard of assessment by most science and engineering instructors.

In order to move instructor practice towards online exams we must show instructors that online exams are at least as effective at measuring student knowledge as traditional on-paper exams. Prior work has shown that carefully designed multiple choice questions can approximate traditionally hand-graded open response problems [2, 6].

This work extends these findings by showing that online quizzes are at least equal to simultaneously administered on-paper quizzes at predicting scores on traditional hand-graded problems on the final exam. Importantly, the online quizzes are significantly better predictors of scores on conceptual questions on that exam.

2. PROCEDURES

To compare online quizzes with traditional testing, we administered weekly quizzes, each comprised of a 25 minute on-paper quiz and a 25 minute online quiz. We compared these quizzes with each other and with other assessments including the final exam. The on-paper quizzes consisted of traditional long-form questions that were graded using a rubric to assign partial credit. The online quizzes were drawn from a comprehensive set of online assessments that our RELATE.MIT.edu group is making, each concentrating on a single topic that would correspond to one week of instruction in a typical introductory Newtonian mechanics course or a single chapter in a typical textbook (e.g. Momentum, Energy, Newton’s Laws, Angular Kinematics, etc.).

To make the online quizzes, we combined questions from research-developed instruments where possible [5, 4] with questions previously used in MOOCs or large on-campus courses for which we were able to calculate Item Response Theory *difficulty* and *discrimination* parameters for each question. Higher discrimination questions yield more information about student ability. The final weekly quiz questions are selected from this corpus based on three criteria: achieving uniform coverage across subtopics, high discrimination and appropriate difficulty, and more quantitative emphasis than typically found on concept tests. We coded questions requiring numerical or symbolic response using the appropriate open response formats.

Two weekly quizzes (weeks seven and nine) were not composed according to this approach. The week seven online quiz consisted of our Angular Procedures Test, an assessment being developed to measure students' ability to perform basic calculations of quantities such as torque, angular momentum, etc., and was all open response; the week nine online quiz consisted of the problem decomposition portion of the Mechanics Reasoning Inventory [3] and was all multiple choice. Both of these correlated with the average of the other tests at the $r \approx 0.1$ level and they were dropped from further analysis.

To account for the variation in mean quiz score from week to week, we calculated z-scores [(deviation from class average)/(standard deviation)] for every student for the on-paper and online portions of each week's quiz.

Many of the online quiz questions allowed several attempts. We suspect *a priori* that a correct answer offered on the first attempt is indicative of greater ability than one given after one or more incorrect attempt. Furthermore, the online quizzes include both multiple choice and open response questions that emphasize conceptual and calculational ability respectively. Consequently, we have explored weighting schemes which award credit differentially according to the number of attempts made and the format of the question.

All of the weighting schemes we studied belong to a three parameter family defined by

$$\begin{aligned} s_{adj}(w_o, P_M, P_O) = \quad & (1 - w_o)\left[(1 - P_M)s_{M,1^{st}} + P_M s_{M,ev.}\right] \\ + \quad & w_o\left[(1 - P_O)s_{O,1^{st}} + P_O s_{O,ev.}\right] \end{aligned} \quad (1)$$

where $s_{M,1^{st}}$ is the score a student earned on their first attempts on the multiple choice questions, $s_{M,ev.}$ is the score earned by the student on multiple choice problems including all attempts, and likewise $s_{O,1^{st}}$ is the first attempt score on open response questions, and so on. The parameter w_o controls the relative weight given to open response and multiple choice problems and P_M and P_O control the amount of partial credit awarded to students who submit correct answers after already having made incorrect attempts on multiple choice and open response questions, respectively. The official grades used in the course were calculated with $w_o = \frac{1}{2}$ (i.e. weighting multiple choice and open response equally) and $P_M = P_O = 1$ (i.e. students got full credit if the ever got the correct answer within the allowed number of attempts).

In the following sections we evaluate the weekly quizzes on the basis of self-consistency and correlation with traditional measures of student knowledge.

		Corr. w/	
Scheme	α	Final	p
$\left(\frac{1}{2}, 0.0, 0.0\right)$	0.74	0.83	0.00012
$\left(\frac{1}{2}, 0.7, 0.7\right)$	0.78	0.86	0.00004
$\left(\frac{1}{2}, 1.0, 1.0\right)$	0.76	0.86	0.00005
$(0.35, 0.4, 0.8)$	0.80	0.88	0.00002

Table 1: Cronbach's α and correlations with final exam scores are presented for weekly on-paper quizzes as well as weekly online quizzes with four different weightings. The notation for the online weighting is a list of the parameters (w_o, P_M, P_O), as defined in Equation 1. Thus, for example, $\left(\frac{1}{2}, 1.0, 1.0\right)$ corresponds to equal weighting of multiple choice and open response with no penalty for multiple attempts. The final line is the scheme which maximizes correlation with the final exam. For the remainder of the paper, we use the parameters in the second line

3. ANALYSIS OF SELF-CONSISTENCY

To quantify the week to week consistency of the written and online tests, we calculated Cronbach's α parameter for both test types. Cronbach's α is a standard statistic used in psychometrics to measure the extent to which a set of items measure the same underlying construct, and is defined as

$$\alpha = \frac{K\bar{c}}{(\bar{v} + (K-1)\bar{c})} \quad (2)$$

where K is the number of items (in this case $K = 11$, the number of weekly quizzes), $\bar{v}$ is the average of the variances of the individual items ($\bar{v} = 1$ in this case because of the use of z-scores), and $\bar{c}$ is the average of all covariances between the items in the assessment. Results are summarized in Table 1 and Fig. 1.

Cronbach's α ranges between zero and one, with zero corresponding to no correlation between items (quizzes). An assessment with an α above 0.7 is considered to have acceptable self-consistency. Hence the value 0.78 observed here indicates good consistency and suggests that there is little systematic dependence of students' ability on the topic of the week.

Of particular note is that all weighting schemes for the online quizzes resulted in better self-consistency than that of the written quiz. Since $1 - \alpha$ is the ratio of the error variance to the observed test variance (which is 1 for z-scores), the online quizzes have at least one third less error than the written quizzes.

4. CORRELATION WITH OTHER ASSESSMENTS

Hestenes' revolutionary Force Concept Inventory [1] showed that success on traditional calculational problems did not imply understanding of concepts. Similar considerations led us to create the Mechanics Reasoning Inventory ("MRI") that tests students' understanding of which physics principles apply in a given situation [3]. We administered the MRI as part of the final exam as well as including additional more general concept questions (CQ). Thus the final

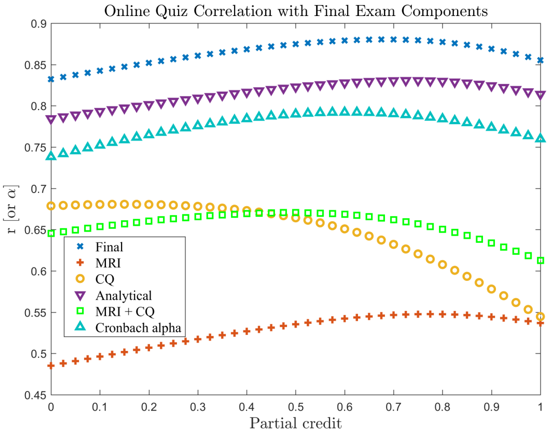


Figure 1: The self-consistency of the online quizzes and their correlation with different components of the final exam are plotted as a function of the amount of credit awarded to correct answers submitted after incorrect answers, independent of problem type. The weightings depicted consist of $s_{adj}(\frac{1}{2}, p, p)$ with p ranging from 0 to 1. A value of $p \approx 0.7$ gives nearly optimal Cronbach's α and correlation with the final exam, so except where otherwise noted this is the value we have used elsewhere in the paper.

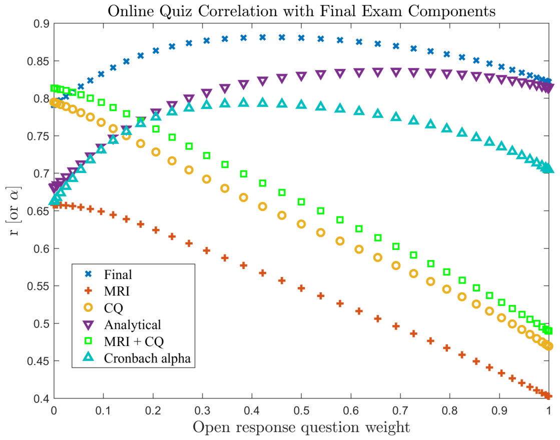


Figure 2: The correlation of the online quizzes with different components of the final exam are plotted as a function of the fractional weight given to open response questions, w_o. Notably, correlation with the concept questions CQ and MRI is best when only MC quiz scores are included, while correlation with the analytical problems on the final exam is best when the OR quiz score is weighted 3:1 over the MC quiz score. The colored horizontal lines indicate the correlation of the written quiz questions with the corresponding exam component.

exam ("FIN") had two types of conceptual questions (MRI and CQ), as well as traditional problems requiring analytical answers.

The multiple choice (MC) and open response (OR) parts of the online online quizzes are mostly conceptual and calculational respectively, and were averaged separately. To explore which parts of the final are best correlated with which type of online quiz questions, we found the correlation coefficients as a function of w_O, the weight of the open response grade vs the MC grade. (See Fig. 2.)

Our most dramatic finding is that all measures of conceptual knowledge correlate best with predictions based *solely* on the MC grade on the online quizzes. Final conceptual knowledge correlates poorly with the online OR quiz scores (0.40), and even worse with with the on-paper quizzes (0.37).

Turning to the traditional symbolic response final questions, we see that the online quiz scores (weighted 3:1 on OR) correlate at 0.83 vs. only 0.78 for the written quiz scores. This is suggestive that the long-form, symbolic, open response written quizzes don't measure anything better than the online quizzes. This finding suggests that the testing and grading error associated with traditional problems outweighs any intrinsic advantage they may have for assessing students. Even if symbolic open-response questions are the "best" indicator of student ability, the online quizzes have better ability to predict this measure because they have less intrinsic error.

5. DISCUSSION AND CONCLUSION

The major limitation of this study is that only N=15 students took the final exam. Thus the statistical errors on the correlation coefficients are typically 0.2 and only a few of our results are statistically significant (at p = 0.05), which we designate with the words 'significant' or 'show'. Nevertheless, we have confidence in other results designated with the word 'suggest'. The basis of our confidence is that some testing error is "common mode" - for example random testing error on the final exam will lower the correlation coefficient of both online and on paper quizzes with the final in a correlated manner, while adding error to both. Also, our results are robust: removing two quizzes, changing the weighting of multiple choice vs. open response, and variations of the partial credit for correct on subsequent attempts do not change the relative performance on online vs on paper quizzes. Repetition of this experiment in larger classes is highly desirable, and is ongoing.

This experiment provides strong evidence that online quizzes out-perform on-paper quizzes containing hand-graded traditional problems in an introductory physics course. It should help educational data miners convince instructors to replace on-paper testing with online testing, which will greatly facilitate studies of learning.

6. ACKNOWLEDGMENTS

We thank the MIT Office of Digital Learning for support of this project.

7. REFERENCES

[1] D. Hestenes, M. Wells, and G. Swackhamer. Force concept inventory. *The Physics Teacher*, 30(3):141–158, 1992.

[2] S.-Y. Lin and C. Singh. Can free-response questions be

approximated by multiple-choice equivalents? *American Journal of Physics*, 81:624–629, 08 2013.

[3] A. Pawl, A. Barrantes, C. Cardamone, S. Rayyan, and D. E. Pritchard. Development of a mechanics reasoning inventory. In *Physics Education Research Conference 2011*, volume 1413 of *PER Conference*, pages 287–290, Omaha, Nebraska, August 3-4 2011.

[4] L. G. Rimoldini and C. Singh. Student understanding of rotational and rolling motion concepts. *Phys. Rev. ST Phys. Educ. Res.*, 1:010102, Oct 2005.

[5] C. Singh and D. Rosengrant. Multiple-choice test of energy and momentum concepts. *American Journal of Physics*, 71(6):607–617, 2003.

[6] B. Wilcox and S. Pollock. Coupled multiple-response vs. free-response conceptual assessment: An example from upper-division physics. 10, 07 2014.

Relation Analysis between Learning Activities on Digital Learning System and Seating Area in Classrooms

Atsushi Shimada
Faculty of Information Science
and Electrical Engineering
Kyushu University
Fukuoka, Japan
atsushi@ait.kyushu-u.ac.jp

Fumiya Okubo
Faculty of Business
Administration
Takachiho University
Tokyo, Japan
fokubo@takachiho.ac.jp

Yuta Taniguchi
Faculty of Information Science
and Electrical Engineering
Kyushu University
Fukuoka, Japan
taniguchi@ait.kyushu-u.ac.jp

Hiroaki Ogata
Academic Center for
Computing and Media Studies
Kyoto University
Kyoto, Japan
hiroaki.ogata@gmail.com

Rin-ichiro Taniguchi
Faculty of Information Science
and Electrical Engineering
Kyushu University
Fukuoka, Japan
rin@ait.kyushu-u.ac.jp

Shin'ichi Konomi
Faculty of Arts and Science
Kyushu University
Fukuoka, Japan
konomi@artsci.kyushu-u.ac.jp

ABSTRACT

This paper discusses a relation analytics between learning activities and seating area in classrooms. Learning activities are collected via digital learning systems; including a learning management system, an e-portfolio system and an e-Book system. The activities are converted into barometers which indicate the amount of activities such as quiz scores, report scores, action frequencies on e-Books, length of journals, etc. The classroom is divided into 12 subareas, and the correspondence between students and the areas are also collected via the learning management system. We applied classical statistical analyses to the collected data. Through the experiments with about 200 students over 14 weeks, we found out that the seating area has strong relationship to learning activities.

Keywords

Learning activities, seating position, classroom

1. INTRODUCTION

Much attention has been paid to learning analytics (LA) and educational data mining (EDM) in recent years, since information and communications technology-based (ICT-based) educational systems have become widespread. Utilizing LA enables us to record various kinds of learning logs. Understanding students' behavior is a crucial issue in LA and EDM research domains. Therefore, there are many studies related to learning behavior analyses, such as behavior clustering[8], learning behavior in programming courses[1], preview and review pattern analyses[4], and academic performance prediction[3]. These studies commonly focus on learning activities corresponding to educational systems (activities in the digital world); they typically do not give much attention to activities in the physical world.

In this study, we focused on face-to-face lectures in which educational systems were introduced, and we analyzed how student seating areas correlated with learning activities recorded in the systems. There are a few related studies discussing the relationship between seating positions and behavior, such as the relationship between seat selection and academic achievement in small classes (less than 35 students)[6], seat location and an analysis of relevant comments from 55 students[7].

In contrast, the focus of our study was a larger scale classroom than in the abovementioned studies. More than 200 students attended the lecture, and the learning activities and student seating areas were examined over 14 weeks. To the best of our knowledge, this is the first study handling a large number of learning activity logs for the relational analysis between learning behavior and seat selections.

2. DIGITAL LEARNING PLATFORM

2.1 M2B system

At Kyushu University in Japan, a digital learning platform, the M2B system, was introduced in 2014. The M2B system consists of three subsystems; a learning management system (Moodle), an e-portfolio system (Mahara), and an e-book system (BookRoll). BookRoll is a self-developed e-book system for providing digital lecture materials and collecting browsing logs.

Various kinds of educational/learning logs are collected by M2B systems. Students submit their reports, answer quizzes, access materials, and reflect on their learning activities using these systems. More precise learning logs are collected by e-book systems (e.g., when a student opens an educational material or when he/she turns a page of the material).

Table 1: Calculation of active learner points (ALPs).

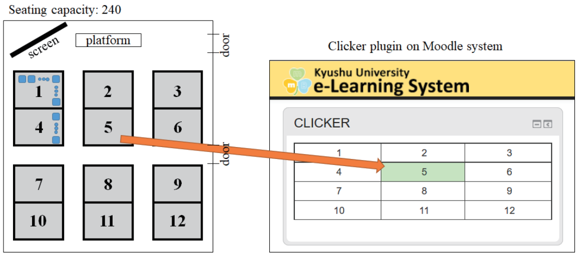
activity	point					
	5	4	3	2	1	0
quiz	Above 80%	Above 60%	Above 40%	Above 20%	Above 10%	Otherwise
report	Submission		Late			Not
login	Upper 10%	Upper 20%	Upper 30%	Upper 40%	Upper 50%	Otherwise
highlight	Upper 10%	Upper 20%	Upper 30%	Upper 40%	Upper 50%	Otherwise
memo	Upper 10%	Upper 20%	Upper 30%	Upper 40%	Upper 50%	Otherwise
action	Upper 10%	Upper 20%	Upper 30%	Upper 40%	Upper 50%	Otherwise
browse	Upper 10%	Upper 20%	Upper 30%	Upper 40%	Upper 50%	Otherwise
diary	Upper 10%	Upper 20%	Upper 30%	Upper 40%	Upper 50%	Otherwise

Table 2: Calculation of activity score during class

activity	point					
	5	4	3	2	1	0
ic_event	Upper 10%	Upper 20%	Upper 30%	Upper 40%	Upper 50%	Otherwise
ic_bookmark	Upper 10%	Upper 20%	Upper 30%	Upper 40%	Upper 50%	Otherwise
ic_highlight	Upper 10%	Upper 20%	Upper 30%	Upper 40%	Upper 50%	Otherwise
ic_memo	Upper 10%	Upper 20%	Upper 30%	Upper 40%	Upper 50%	Otherwise

2.2 Collection of Seating Area

Although the Moodle system has a plug-in that manages attendance of students, it cannot record seat positions. Another possibility is a student attendance system based on face detection[2]. However, the face detection technique is not perfect, so correct seat positions cannot be identified. Therefore, we developed a clicker system as a plug-in for the Moodle. Usually, the clicker is used for collecting answers from students. In our study, we utilized the clicker plug-in to collect information on the seating areas in the classroom. At the beginning of the weekly class, a teacher asked students to identify their seating areas by clicking the corresponding area number. As shown in Fig. 1, the classroom has about 240 seats, and the area is divided into 12 subareas.



1	2	3
4	5	6
7	8	9
10	11	12

Figure 1: Left: top view of the classroom. About 240 seats are available in the classroom. The classroom is divided into 12 areas to collect the seating area of students. Right: clicker plugin on Moodle system. Students answer their seat area by clicking the corresponding area number.

2.3 Active Learner Point: ALP

We utilized Active Learner Points (ALPs) as barometers of learning activities calculated from various kinds of logs stored in the M2B system. In this study, we utilized three activities (quizzes, reports, and logins), four activities (highlight, memo, action, and browse), and an activity involving diary length from the Moodle system, e-book system, and Mahara system, respectively. Each activity was evaluated by the students using a conversion to a 5-level scoring system, as summarized in Table 1. Please refer to the literature[5] for a more detailed explanation about ALPs.

2.4 Learning Activities during On-site Class

The abovementioned ALPs mainly reflect students' out-of-class activities, such as previewing and reviewing. To analyze activities taking place in on-site classes, we introduced new scores calculated from e-book operation logs. The calculation of scores was inspired by the ALP, as shown in Table 2. To distinguish the scores from those of ALPs, we used the word "ic_" to indicate "in-class" activities. The score reflects the frequency of usage in each operation: how many times a student operates the e-book (ic_event), how often a student uses the bookmark operation, and students' usage of highlight and memo functions (ic_bookmark, ic_highlight, and ic_memo, respectively).

3. DATASET

We collected learning activity logs (in fact, more than 890,000 records in the database) over 14 weeks from a course in information science conducted at our university. This course is designed to provide an introduction to ICT technology in a number of disciplines. The course consists of a series of sessions on the major research areas of this technology, including an initial discussion of the conceptual foundations of algorithms, image processing, and character recognition.

About 200 students attended the classes every week. Data regarding learning activities during classes (i.e., in-class activities) and activities outside of class, such as the preview/review of materials, were collected through the M2B system. At the beginning of each class, the students identified their seating positions using the clicker plug-in. During a 90-minute lecture, the students opened the e-book and followed the explanation therein while creating bookmarks, highlighting texts, and creating memos as necessary.

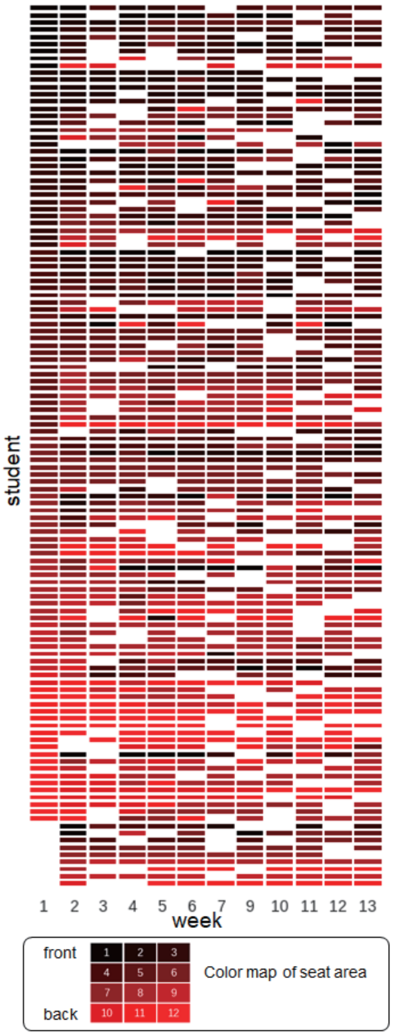


Figure 2: Area transition over 12 weeks. The 8th week and 14 week are removed because of examination weeks. The row corresponds to each student. From top to down, and from 1st week to 13 week, the seating area is sorted by the area number.

4. ANALYTICS RESULTS

4.1 Transition of Seating Area

We analyzed the seating areas by visualizing the transition over weeks. The classes were conducted 14 times, but we excluded the 8th and 14th weeks when students were taking examinations. To avoid sparse visualization, we collected data on students who attended more than eight of 12 weeks.

Fig. 2 is the visualized result of the transition in seating areas. The horizontal axis and vertical axis represent the i-th week and individual student, respectively. Therefore, a single row refers to a student's seating area transition(s). The color of each cell corresponds to the color map shown in the bottom part of Fig. 2. From the darker to brighter color, the seating area from #1 to #12 is represented. From the first column to the last column, the seating area is arranged in ascending order, corresponding to individual students.

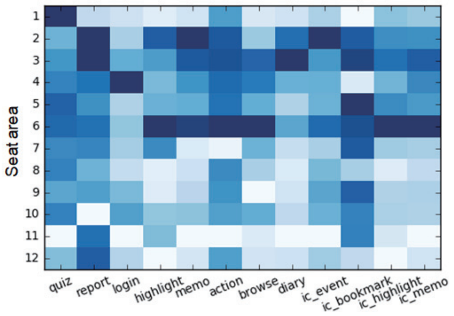


Figure 3: Distribution of learning activity scores. The horizontal axis is the item of activity score, and the vertical axis indicates the seating area. The darker color, the score is higher.

Table 3: t-test results. *: p<0.05, **: p<0.01

	front		back		
item	ave	std	ave	std	p
quiz	4.89	0.75	4.86	0.51	
report	2.93	0.88	2.95	0.81	
login	1.71	1.05	1.45	1.08	
highlight	0.46	0.94	0.13	0.34	*
memo	0.65	1.01	0.20	0.41	**
action	4.18	0.98	3.75	1.10	*
text	2.28	1.22	1.82	1.17	*
diary	1.02	1.26	0.43	0.57	**
ic_event	1.69	1.91	0.89	1.46	*
ic_bookmark	3.39	0.71	3.27	0.52	
ic_highlight	2.82	1.12	2.34	0.73	**
ic_memo	2.41	1.61	1.54	1.00	**
attend	8.98	3.53	7.69	3.24	*

Fig. 2 suggests that most students did not change seating areas often. They remained in the same area or one close by over several weeks. On the other hand, some students changed seating areas frequently (every week). Such students were more likely to be absent from classes. To investigate this subject quantitatively, we evaluated the variety of seating areas by comparing two student groups. In one group, members attended classes for nine or more weeks (123 students). In the other group, members attended classes for less than nine weeks but for at least three weeks (79 students). Note that we excluded students who attended class for less than three weeks to avoid meaningless calculations of seating area variations. For individual students in each group, we initially calculated the standard deviation (SD) of the seating area and then evaluated the average of SDs. The averages were 1.39 for the former group and 1.74 for the latter group, respectively. There was a significant difference ($p < 0.05$) between the two groups.

4.2 Front Area versus Back Area

We analyzed the respective scores for 12 types of learning activities: eight types of ALPs and four types of on-site class activities. Fig. 3 shows the seating area versus the item matrix (12 × 12 matrix) and represents the distribution of scores. The matrix element in blue corresponds to the score (the darker the color, the higher the score).

Overall, we can see that scores in the front areas (from #1 to #3) are higher than those in the back areas (from #10 to #12). This result suggests a hypothesis that students seated toward the front of the classroom participated in more activities than those in the back of the classroom. To investigate the hypothesis, we conducted a t-test for each item and assimilated the results as shown in table 3. Significant differences between groups were noted for nine of 12 items. Most differences originated from the activities related to e-book operations. Regardless of in-class/out-of-class activities, students seated toward the front of the classroom tended to engage in many activities accessible through the e-book system. Considering that the scores were based on the frequency of reviewing/previewing activities, we can summarize that students seated in the front of the classroom tended to perform these activities. In addition, these students utilized the e-book during class.

Regarding the remaining four items (i.e., quizzes, reports, login function, and ic_bookmark function), there was no significant difference between groups. Intuitively, people tend to think that students seated toward the front of a classroom get higher scores on quizzes; however, a significant difference was not identified in our experiments. The score of "report" indicated whether a student submitted his/her report, not the quality of the report itself. Therefore, most students earned similar scores. With regard to the login and ic_bookmark functions, there were fewer learning logs, which were not sufficient for performing statistical analyses.

5. CONCLUSION

In this study, we analyzed the relationship between learning activities and seating areas in classrooms. The learning activities were collected by the digital learning platform over 14 weeks and converted to scores, which indicated the amount of activities of the 12 items. Information regarding seating area was collected via a clicker plug-in on the Moodle system. We conducted t-tests to perform individual item analytics.

Overall, we found out that students with higher learning activity scores tended to sit toward the front of the classroom. From the seating area transition shown in Fig. 2, most students sat in the same area over weeks. This fact implicitly suggests that students who are highly motivated tend to select seats in the front of the classroom rather than in other areas. Furthermore, students' selections of seating areas did not change drastically over several weeks.

However, our current analysis has a limitation in that we could not investigate the motivations of students. Therefore, in our future work, we will analyze the relationship between the motivations and activities of students. Furthermore, the current strategy of collecting data regarding seating areas should be improved to grasp whether a student selects a seating area aggressively or passively. For example, a student who is late to class will be forced to select a seat near the front of the classroom because other areas are already fully occupied. To address the issue, we will ask students to select the seating area as soon as possible upon entering the classroom. A timestamp analysis would be helpful for grasping the situations of students.

In our future work, we will continue with the analytics of data collected in other classrooms and investigate whether the conclusions of this study can be applied generally to other classes and courses. Furthermore, we will introduce a new criterion such as self-efficacy[9] for the analytics involving seat selection and the motivation of students.

Acknowledgements

This work was supported by JST PRESTO Grant Number JPMJPR1505, and JSPS KAKENHI Grand Number JP16H06304, Japan.

6. REFERENCES

[1] X. Fu, A. Shimada, H. Ogata, Y. Taniguchi, and D. Suehiro. Real-time learning analytics for c programming language courses. In Proceedings of the Seventh International Learning Analytics & Knowledge Conference, pages 280–288, 2017.

[2] S. Lukas, A. R. Mitra, R. I. Desanti, and D. Krisnadi. Student attendance system in classroom using face recognition technique. In 2016 International Conference on Information and Communication Technology Convergence (ICTC), pages 1032–1035, 2016.

[3] K. Mouri, F. Okubo, A. Shimada, and H. Ogata. Bayesian network for predicting students' final grade using e-book logs in university education. In IEEE International Conference on Advanced Learning Technologies(ICALT2016), pages 85–89, 2016.

[4] M. Oi, F. Okubo, A. Shimada, C. Yin, and H. Ogata. Analysis of preview and review patterns in undergraduates' e-book logs. In The 23rd International Conference on Computers in Education (ICCE2015), pages 166–171, 2015.

[5] F. Okubo, T. Yamashita, A. Shimada, and S. Konomi. Students' performance prediction using data of multiple courses by recurrent neural network. In 25th International Conference on Computers in Education (ICCE2017), pages 439–444, 2017.

[6] B. A. V. Schee. Marketing classroom spaces: Is it really better at the front? Marketing Education Review, 21(3):191–200, 2011.

[7] P. Tory, H. Olivia, and E. Dennis. The effect of seat location and movement or permanence on student-initiated participation. College Teaching, 59(2):79–84, 2011.

[8] G. Wang, X. Zhang, S. Tang, H. Zheng, and B. Y. Zhao. Unsupervised clickstream clustering for user behavior analysis. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, CHI '16, pages 225–236, 2016.

[9] M. Yamada, C. Yin, A. Shimada, K. Kojima, F. Okubo, and H. Ogata. Preliminary research on self-regulated learning and learning logs in a ubiquitus learning environment. In 15th IEEE International Conference on Advanced Learning Technologies, ICALT 2015, Hualien, Taiwan, July 6-9, 2015, pages 93–95, 2015.

Qmatrix-generated Autoencoder: Automatic Mapping Question Items to Skills

Pan Liao
School of Computer and
Control Engineering
University of Chinese
Academy of Sciences, China
liaopan15@mails.ucas.ac.cn

Yuan Sun
Information and Society
Research Division
National Institute of
Informatics, Tokyo, Japan
yuan@nii.ac.jp

Shiwei Ye
School of Computer and
Control Engineering
University of Chinese
Academy of Sciences, China
shiwye@ucas.ac.cn

Guiping Su
School of Computer and
Control Engineering
University of Chinese
Academy of Sciences, China
sugp@ucas.ac.cn

Junyi Dai
School of Computer and
Control Engineering
University of Chinese
Academy of Sciences, China
daijunyi16@mails.ucas.ac.cn

Yi Sun
School of Computer and
Control Engineering
University of Chinese
Academy of Sciences, China
sunyi@ucas.ac.cn

ABSTRACT

Computer-aided assessment and online intelligent tutoring systems have great potential in assessing student skills in order to tailor course contents and adaptively provide customized hints, when students meet a bottleneck. However, these systems rely on the mapping of items to skills; this is called Q-matrix. The construction of the Q-matrix is a labor-intensive hand-engineered task; therefore, the demand for automatically constructing the Q-matrix for online assessment platforms is ever critical. To address this problem, we utilize the autoencoder, which is an artificial neural network used for unsupervised learning in the effective learning of robust dataset representation. We propose a Q-matrix-generated autoencoder (QAE) model, as an approach to automatically learn the Q-matrix from unlabeled data based on skill constraint. Comparative experiments, on an artificial and two real-world datasets, clearly show a promising result when using the QAE to construct the Q-matrix. Moreover, state-of-the-art performance was achieved when constructing the Q-matrix on the basis of reconstructing the original data.

Keywords

the mapping of items to skills; Q-matrix; Q-matrix-generated autoencoder model

1. INTRODUCTION

In the field of educational data mining, the cognitive diagnostic models require the mapping of items to skills (i.e., Q-matrix) in order to determine the skills mastered by the student. Intuitively speaking, the Q-matrix is a binary representation illustrating the relationship between test questions and the learner's latent traits[1]. However, it is difficult and tedious to analyze which skills are involved in an item. Therefore, the demand for the automatic construction of the Q-matrix, for online assessment platforms, is ever critical and a more effective method to construct the Q-matrix is needed. Therefore, a more effective method for automatically building the Q-matrix is needed. With the growing popularity of deep learning methods, we attempt to utilize the autoencoder method in order to deal with the hard problem of constructing the Q-matrix. In this paper, the main contribution of our work is to propose the QAE model, which can automatically learn the Q-matrix from student response data, while also being able to reconstruct the student response data, even though with some noise included.

2. RELATED WORK

2.1 Autoencoder

An autoencoder is a neural network trained in attempting to copy its input to its output used in the unsupervised learning of efficient input coding. The aim of the autoencoder is to learn an effective and robust representation for a data set. In addition, it can also be used in dimensionality reduction. An autoencoder[6] consists of the encoder and the decoder, which can be defined as the deterministic mapping f_θ and g_θ. An autoencoder takes an input $\mathbf{x} \in [0,1]^d$ and maps it to a hidden representation $\mathbf{y} \in [0,1]^{d'}$ through f_θ, such that: $\mathbf{y} = f_\theta(\mathbf{x}) = s(\mathbf{W}\mathbf{x} + \mathbf{b})$ where s is a non-linearity such as the sigmoid and $\theta = \{\mathbf{W}, \mathbf{b}\}$. $\mathbf{W}$ is a $d' \times d$ weight matrix and $\mathbf{b}$ is a bias vector.

3. QMATRIX-GENERATED AUTOENCODER ARCHITECTURE

Fig.1 illustrates the structure of our QAE model. The model obtains two parts, the upper part is where we use the Q-matrix to constrain the hidden representation, called skill constraint, in the training process only. For example, for the item i input data $\mathbf{r}_i$, the QAE model can obtain the latent representation $\mathbf{y}_i$ with $\mathbf{q}_i$ constrained. The other part is a basic autoencoder with corrupted input, and $L(\mathbf{r}, \mathbf{z}) + L(\mathbf{q}, \mathbf{y})$ denotes the loss function for $(\mathbf{r}, \mathbf{z})$ and $(\mathbf{q}, \mathbf{y})$. In order to improve the model's performance, we used cross-entropy as a loss function with an added trade-off parameter β; then, the loss function can be rewritten as $\beta \cdot L_H(\mathbf{r}, \mathbf{z}) + (1 - \beta) \cdot L_H(\mathbf{q}, \mathbf{y})$. And the structure of our Deep-QAE model is similar to the QAE model. The only difference in the two models is that the Deep-QAE uses a stacked autoencoder model consisting of multiple layers. Thereby, it is able to obtain a richer and more effective hidden representation when carrying out various tasks.

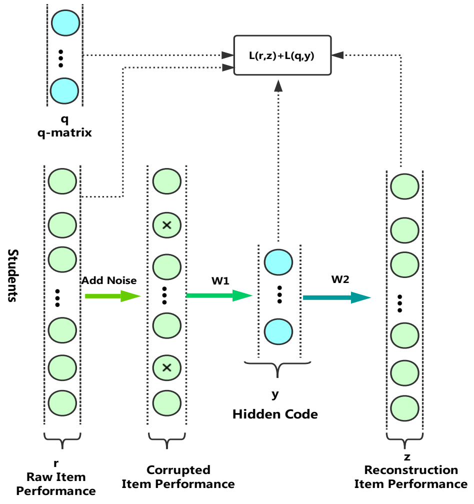


Figure 1: Illustration of QAE model

Regularization is a very important technique for preventing overfitting. In our model, we added an L1 regularization term to the hidden representation $\mathbf{y}$ and an L2 regularization term to the weight matrices $\mathbf{W}$. After training QAE model with Q-matrix constraint, our model can predict the new Q-matrix from the new student response data. Because we only added the Q-matrix constraint during training, we did not add the Q-matrix to it during the prediction stage. This is a typical way of training deep learning models using labled data to train the model, and then using the model to predict new data labels.

4. EXPERIMENTAL EVALUATION

4.1 Datasets Introduction

In this study, we use both artificial and real datasets in order to evaluate the performance of the algorithm. We divided the experiment into two parts: (i) artificial dataset, and (ii) real dataset experiments. This makes it easier to validate whether or not the algorithms succeeded in achieving accurate results. To generate the artificial data, we model the learner's latent traits and the expert Q-matrix, in order to generate the ideal leaner response as denoted by IDR [5]. This allows us to obtain an accurate measure of learner performance. With regard to real datasets, we use two datasets consisting of fraction-subtraction data($FrcSub$[1]) and Japanese language learning data(CAS), respectively. The private dataset CAS was collected from Japanese language learning courses for undergraduates in 2013, and it includes 73 problems and 14 skills.

4.2 Experimental Results

4.2.1 Evaluation Criteria

In the experiment, we used two popular prediction metrics to evaluate the algorithms for each dataset; namely, Accuracy (Acc.), and Root Mean Squared Error (RMSE). In order to verify the performance of our models, we used baseline approaches to compare. BMC[7]: A latent space perturbation algorithm for Boolean matrix completion, based on weighted Frobenius Norm, for predicting the missing values in a binary matrix of an educational area. NMF[3]: Non-negative matrix factorization is a useful decomposition technique for multivariate data, with the property of all matrices having no negative elements. It can be used for dimensionality reduction or topic extraction. MLP[4]: A multilayer perceptron is a feedforward artificial neural network model, with each layer fully connected to the next one. AE[2]: An artificial neural network used for unsupervised learning in the effective learning of a robust dataset representation.

To observe model performance, we constructed training sets of different sizes, with 30%, 60%, 80%, and 95% of student response data for each item. We selected the training set randomly, from the original dataset, and used the remaining data as the test set. We repeat the evaluation 10 times, with different randomly selected training sets, and obtained the average RMSE and Acc. evaluation metrics. We added a dropout layer in our QAE model, and used a fixed dropout rate of 0.5 in order to achieve adaptive regularization during model training. The Deep-QAE network architecture has an architecture of 'I-K-K-K-K-I' for the datasets. I denotes the number of students and K denotes the number of skills. QAE and Deep-QAE models are both implemented by Keras on an Nvidia Tesla K80 GPU with 12 GB of memory. For the purpose of comparison, we recorded the best performance of each algorithm by tuning the parameters. NMF-K denotes that it has K latent factors corresponding to the same number of skills in the Q-matrix. For the MLP method, we used student response data as features, and used the Q-matrix for labelling. Here, the AE model is a basic autoencoder model, with the same network structure as our QAE model. Moreover, it handles the student response data without skill constraint.

Table 1 shows the average RMSE of BMC, NMF, MLP, AE, QAE and Deep-QAE, with different percentages of training data in the three datasets, in Q-matrix generation. Our models outperform other algorithms over all three datasets, with different training percentages, in addition to obtaining lower RMSE. For instance, when the training data is greater than 60%, our models perform better than other algorithms. More importantly, with the increase in skill delineation complexity with regard to the question items, it becomes more difficult for experts in related fields of knowledge to obtain the accurate Q-matrix. For instance, for the $FrcSub$ dataset, the questions are intended for primary school students; therefore, they can easily be defined by human experts. However, in the CAS dataset, the Japanese learning questions are designed for undergraduates; therefore, defining them is a more complicated and difficult task. Then, with the increase in the complexity of the real-world Q-matrix, the BMC, AE, NMF-K models do not fit the data sets very well and cannot obtain good results for Q-matrix generation. Consequently, the MLP algorithm's performance is barely satisfactory.

Table 2 shows accuracy results, which compare NMF-K, BMC, AE, MLP, QAE, and $Deep$-QAE. We found that

[1] https://cran.r-project.org/web/packages/CDM/CDM.pdf

Table 1: RMSE of the experimental Q-matrix generation results

Model	SimulatedData				FrcSub				CAS			
	30%	60%	80%	95%	30%	60%	80%	95%	30%	60%	80%	95%
NMF-K	0.7259	0.7225	0.7249	0.7249	0.6101	0.6212	0.6267	0.6283	0.4823	0.5005	0.5207	0.5368
BMC	0.6597	0.6780	0.6789	0.6863	0.6020	0.5968	0.5809	0.5916	0.5683	0.4681	0.4713	0.4713
AE	0.6702	0.6605	0.6607	0.6469	0.7062	0.6960	0.7199	0.7490	0.7009	0.8018	0.7724	0.7432
MLP	0.5098	0.4783	0.4549	0.4480	0.4089	0.3865	0.4021	0.2798	0.4550	0.4437	0.4324	0.4244
QAE	**0.500**	0.4997	0.4470	**0.3925**	0.4720	0.4211	0.4424	0.3560	0.5115	0.4722	0.4310	0.4085
Deep-QAE	0.5262	**0.4767**	**0.4465**	0.4085	**0.4038**	**0.3817**	**0.3883**	**0.2414**	**0.4549**	**0.4397**	**0.4074**	**0.4012**

Table 2: Accuracy of experimental Q-matrix generated results

Model	SimulatedData				FrcSub				CAS			
	30%	60%	80%	95%	30%	60%	80%	95%	30%	60%	80%	95%
NMF-K	0.4731	0.4780	0.4745	0.4745	0.6275	0.6138	0.6069	0.6050	0.7674	0.7494	0.7289	0.7118
BMC	0.5325	0.5300	0.5313	0.5275	0.6375	0.6438	0.6625	0.6500	0.6808	0.7808	0.7779	0.7779
AE	0.5502	0.5628	0.5619	0.5771	0.4920	0.5125	0.4783	0.4250	0.5033	0.3567	0.4024	0.4446
MLP	0.7089	0.7331	0.7600	0.7583	0.8125	0.8297	0.8156	0.8625	0.7732	0.7757	0.7800	0.7804
QAE	**0.7489**	0.7496	**0.7987**	**0.8292**	0.7759	0.8187	0.8000	0.8375	0.7379	0.7760	0.8129	**0.8304**
Deep-QAE	0.7227	**0.7719**	0.7975	0.8083	**0.8170**	**0.8359**	**0.8313**	**0.9000**	**0.7747**	**0.7974**	**0.8329**	**0.8357**

the prediction results from these datasets provide a more stable prediction accuracy in Q-matrix generation. This indicates that our QAE models perform better in the automatic learning of the Q-matrix from unlabeled real-world data based on skill constraint, while simultaneously reconstructing the original dataset by using the reconstruction properties of autoencoder model.

Specifically, in the *CAS* dataset, our models can achieve best performance for more complex, real-world datasets. This proves that our algorithms are better at modeling the hidden knowledge skills of question items. This demonstrates their potential ability to capture item characteristics more precisely, in real-world scenarios, where the questions are very complicated, and the response data is very sparse and include noise. In summary, we demonstrated that our model can autonomously learn the latent knowledge skills of question items with greater precision, even if the original response data include some noise. Therefore, our method is more suitable to real-world testing scenarios, where the datasets are complicated, sparse, and contaminated.

5. CONCLUSIONS

This study was motivated by the need to develop good training algorithms for automatically mapping question items to skills. We overcame this problem by studying the autoencoder applied to the Q-matrix generated task. In this paper, we introduced a very simple and effective model called QAE. Furthermore, we described our QAE architecture for Q-matrix generation. Our QAE model can effectively learn the Q-matrix and reconstruct the original student response data with noise. Our experimental results show that our QAE method provides a promising means to generating a Q-matrix with better delineated skills, for the question items obtained from student response data. However, the process of Q-matrix generation is very complex; therefore, in future work, we will need to discover a better QAE network structure in order to capture the items' latent traits, under certain conditions, with a higher degree of accuracy.

6. ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Numbers 18005852.

7. REFERENCES

[1] T. Barnes. The q-matrix method: Mining student response data for knowledge. In *American Association for Artificial Intelligence 2005 Educational Data Mining Workshop*, pages 1–8, 2005.

[2] Y. Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[3] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[4] R. Reed and R. J. Marks. *Neural smithing: supervised learning in feedforward artificial neural networks.* Mit Press, 1999.

[5] Y. Sun, S. Ye, G. Su, and Y. Sun. Q-matrix learning and dina model parameter estimation. In *Behavioral, Economic and Socio-cultural Computing (BESC), 2016 International Conference on*, pages 1–6. IEEE, 2016.

[6] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[7] H. Wang, G. Su, Y. Sun, S. Ye, P. Liao, and Y. Sun. A latent space perturbation algorithm for boolean matrix completion based on weighted frobenius norm. In *Behavioral, Economic and Socio-cultural Computing (BESC), 2015 International Conference on*, pages 43–46. IEEE, 2015.

Discovering Hidden Browsing Patterns Using Non-Negative Matrix Factorization

Kousuke Mouri
Tokyo University of agriculture and Technology
Tokyo, Japan
mourikousuke@go.tuat.ac.jp

Atsushi Shimada
Kyushu University
Fukuoka, Japan
atsushi@ait.kyushu-u.ac.jp

Chengjiu Yin
Kobe University
Kobe, japan
yin@lion.kobe-u.ac.jp

Keiichi Kaneko
Tokyo University of agriculture and Technology
Tokyo, Japan
k1kaneko@cc.tuat.ac.jp

ABSTRACT

So far, educational researchers have focused on analyzing and visualizing and mining digital textbook logs to enhance the quality of teaching and learning. Previous digital textbook works did not collect the data which positions of the pages learners were browsing in digital textbooks unless eye-tracking technologies. This paper proposes a method to collect the data which positions of the pages learners were browsing in the digital textbooks. To discover various learning behaviors from digital textbook logs collected by our method, this study adopts non-negative matrix factorization technique. We used 50-page browsing and 867-block browsing logs of 36 students, and discovered five kinds of browsing patterns.

Keywords

Digital textbook, non-negative matrix factorization, data mining.

1. INTRODUCTION

A digital textbook has been known as an e-textbook, electronic book and e-book. In recent years, the digital textbook technologies have been introduced to schools and universities in many countries [1], [2], [3]. Japan governments announced to introduce the digital textbooks into all K12 schools by 2020 [4]. Majorities of countries' digital textbook policies only focus on introducing the digital technology. In the digital textbook studies, researchers suggested that introducing digital textbook technologies lead to enhance the learning efficacy and quality of education.

Recently, a few researchers have focused on analyzing, visualizing and mining the digital textbook logs in order to find the following points [5], [6], [7], [8]: (1) learning materials to be improved, (2) learning processes and learning patterns, (3) students' comprehensive level. In analyzing digital textbook logs, they use the data which pages the learners were browsing in the digital textbooks. But, they did not consider the data which positions of the pages the learners were browsing in the digital textbooks. It is difficult to collect the data unless using eye-tracking technologies. In addition, it is difficult to give eye-tracking equipment for all students because eye-tracking equipment is a little expensive.

To tackle the issue, this study proposes a method to collect the data. By analyzing the logs by our method, it enables teachers to grasp which positions of the pages learners were browsing in the digital textbooks.

However, it is insufficient to understand behaviors of learners because of their diversity and high dimensionality. To discover their various learning behaviors, this paper adopts non-negative matrix factorization (NMF) technique [11], which is known as akin to principal component analysis. This study analyzes to discover their browsing patterns based on two methods: (1) which pages learners were browsing in the digital textbooks and (2) which positions of the pages learners were browsing in the digital textbooks.

2. DIGITAL TEXTBOOK SYSTEM

To collect more detailed data about digital textbooks, this study developed a digital textbook system called SEA (Smart E-textbook Application)-Reader. Figure 1 shows the interface of SEA-Reader.

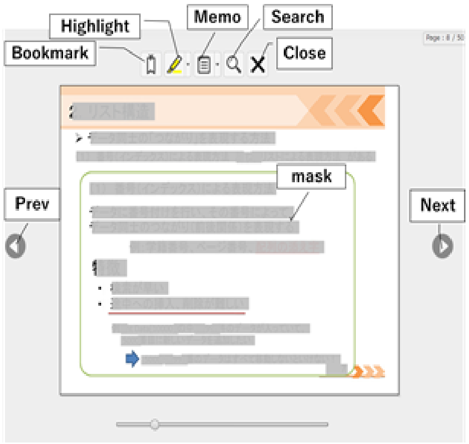


Figure 1. SEA-Reader interface

By using the system, learners can use several functions such as next, prev, bookmark, highlight and memo. Unlike previous digital textbook works [9], [10], the system automatically hides the texts in the digital textbooks with mask processing before the learners browse the texts in the digital textbooks. For example, if there are five text areas in a page as shown in Figure 2, the system covers the text areas in each row with mask processing.

3. METHOD

To discover several browsing patterns, this study uses non-negative factorization (NMF). NMF approximately decomposes a matrix of $n \times m$ positive numbers V as product of two matrices:

$$V \approx WH \qquad (1)$$

According to Shimada et al. [12], they reported that the matrix V named "browsing matrix" is represented by the fact whether a student browsed a page or not. More specifically, they set an element $V_{i,j}$ of the matrix V by

$$V_{i,j} \begin{cases} 1 & (if\ t_{i,j} > th) \\ 0 & (otherwise) \end{cases} \qquad (2)$$

where $t_{i,j}$ is the duration of page i browsed by student j. The decomposed matrices represent two latent relationships: "page browse vs. patterns" given by matrix W and "patterns vs. students" given by matrix H.

Unlike their study, this study represents that the matrix V named "masked browsing matrix" is represented by the fact whether a student browsed a masked text area in a page or not. We set an element $t_{i,j}$ is the duration of a masked text area i browsed by student j. The decomposed matrices represent two latent relationships: "masked block browse vs. patterns" given by matrix W and "patterns vs. students" given by matrix H

4. EXPERIMENTS

4.1 Instruments

The browsing matrix and masked browsing matrix were created from 36 first-year students in a programming education course at Tokyo University of Agriculture and Technology in Japan. The students were required to preview a digital textbook in advance before the lecture. Table 1 shows the details of the digital textbook. The digital textbook consists of five sections. The first section describes the contents of previous lecture regarding an array pointer. Second section describes a list structure. Third section describes an example program of the list structure. Fourth section describes an exercise regarding the list structure. Final section describes hints to complete the exercise.

The digital textbook consists of 50 pages and masked 867 blocks. Therefore, the V of the page browsing pattern is represented by 36-row × 50-column matrix as shown in Figure 2, while the V of masked browsing pattern is 36-row × 867-column matrix as shown in Figure 3. NMF was performed to find five patterns based on two matrices: "browsing matrix" and "masked browsing matrix".

Table 1. The Structure of the digital textbook

	Detail
Page 1~7	The contents of previous lecture
Page 8~19	A list structure
Page 20~30	An example program of the list structure
Page 31~36	An exercise regarding the list structure
Page 37~50	Hints of the exercise

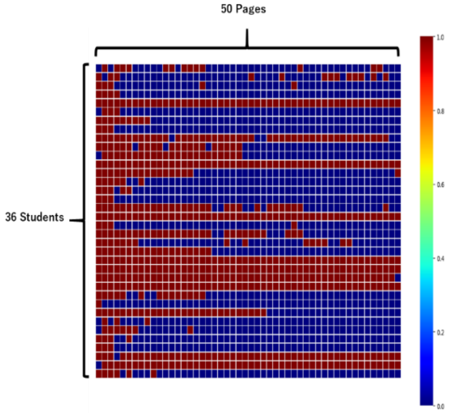


Figure 2. browsing matrix (36-row × 50-column). The red and blue color show the value of $V_{i,j}$, red for one, blue for zero, where the *th* was set to be 10 seconds

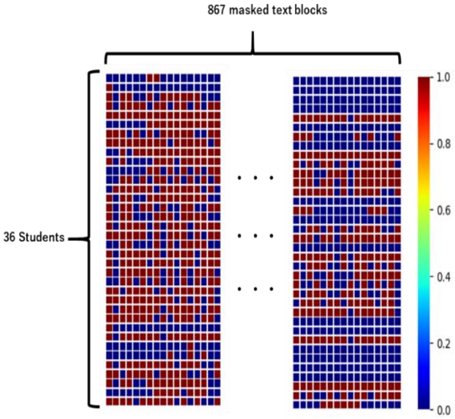


Figure 3. masked browsing matrix (36-row × 867-column). The red and blue color show the value of $V_{i,j}$, red for one, blue for zero, where the *th* was set to be 5 seconds

4.2 Results

4.2.1 *Browsing matrix*

Figure 4 shows the decomposed browsing matrix W and Figure 5 shows the decomposed browsing matrix H. The color scale is decided from green (low) to red (high). Each pattern can be roughly described as Table 2.

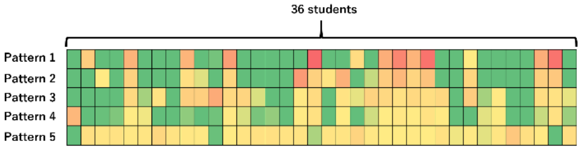


Figure 4. Visualized browsing matrix W

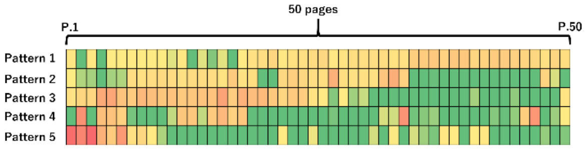


Figure 5. Visualized browsing matrix *H*

Table 2. Discovered browsing patterns

	Detail
Pattern 1	Browse the whole pages
Pattern 2	Browse the pages from 1 to 36
Pattern 3	Browse the pages from 1 to 30
Pattern 4	Browse the pages from 1 to 19
Pattern 5	Browse the pages from 1 to 7

After the NMF, this study summarized the details of students' learning achievements based on each learning pattern as shown in Table 3. The average score and the standard deviation mean the scores calculated from top 10 students in each learning pattern.

Table 3. Details of students' learning achievement based on each learning pattern (Browsing matrix)

	Attendance (Scoring 14)		Exercise (Scoring 10)		Report (Scoring 20)	
	AVG.	SD.	AVG.	SD.	AVG.	SD.
Pattern 1	13.3	1.18	9.6	0.8	19.5	0.67
Pattern 2	13.3	1.18	8.6	2.01	18.6	2.91
Pattern 3	13.9	0.3	8.4	2.49	17.5	3.07
Pattern 4	13.3	0.9	8.1	3.28	17.7	3.92
Pattern 5	13.5	0.92	8	3.09	17.6	3.37

We compared each pattern with attendance, exercise and report scores. The exercise was conducted after a teacher explained about the contents of the digital textbook. The average score of the exercise of the pattern 5 was lower than other patterns, while pattern 1 got the highest score. The average score of the report of the pattern 1 was higher than other patterns. We guess that the students in the pattern 1 were able to enhance their understanding because they previewed whole pages in the digital textbook well. On the other hand, the students in pattern 5 were not able to enhance their understanding because they only previewed the former parts in the pages.

4.2.2 Masked browsing matrix

Figure 6 shows the decomposed mask browsing matrix *W* and Figure 7 shows the decomposed mask browsing matrix *H*. The color scale is decided from green (low) to red (high). Each pattern can be roughly described as Table 4.

Table 4. Discovered masked browsing patterns

	Detail
Pattern 1	Browse the whole blocks
Pattern 2	Browse blocks from the former to middle parts in the whole pages
Pattern 3	Browse whole blocks in the pages from 1 to 17 and from 20 to 30
Pattern 4	Browse whole blocks in the pages from 1 to 19
Pattern 5	Browse whole blocks in the pages from 1 to 7

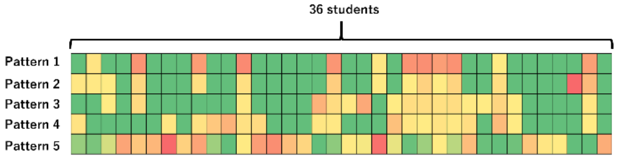


Figure 6. Visualized masked browsing matrix *W*

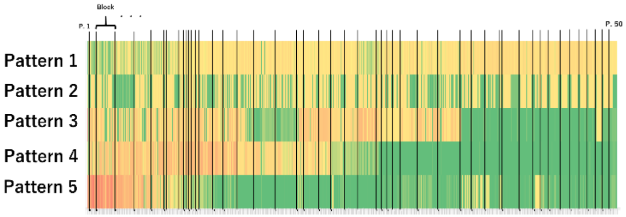


Figure 7. Visualized masked browsing matrix *H*

Table 6 shows the details of students' learning achievements based on the masked browsing matrix. The average score and standard deviation mean that the scores calculated from top 10 students in each learning pattern.

We compared each pattern with attendance, exercise and report scores. The average score of the pattern 1 in the both exercise and report were higher than other patterns. We guess that the students in the pattern 1 were very diligent because they clicked whole blocks to preview the contents of the digital textbook. On the other hand, the average score of the pattern 2 in both exercise and report were lower than the pattern 1. We guess that the students in the pattern 2 were not able to enhance their understanding because they did not click the masked blocks in the digital textbook well.

The average scores of the pattern 3, 4 and 5 were lower than pattern 1. The students in the pattern 3, 4 and 5 did not previewed the contents from page 31 to 50. The contents of these pages were related to the exercise and report to enhance their understanding. Therefore, they did not get better exercise and report scores than the pattern 1.

Table 6. The details of students' learning achievement based on each learning pattern (Masked browsing matrix)

	Attendance (Scoring 14)		Exercise (Scoring 10)		Report (Scoring 20)	
	AVG.	SD.	AVG.	SD.	AVG.	SD.
Pattern 1	13.7	0.45	9.2	1.83	18.2	2.89
Pattern 2	13	1.34	7.4	3.46	17.4	3.9
Pattern 3	13.7	0.45	8	2.36	17.5	4.94
Pattern 4	13.5	0.92	8	3.34	18.1	1.77
Pattern 5	13.7	0.45	7.8	3.09	17.6	2.08

4.3 Discussion

By decomposing the browsing matrix and masked browsing matrix, this study discovered five patterns as shown in table 3 and 5. The results of the browsing matrix indicated that students got the highest exercise and report scores because they previewed whole pages in the digital textbook well. However, we were not able to find the fact of whether students really previewed the contents in the pages if using the browsing matrix.

Therefore, we found that the fact of whether they really browsed the contents in the pages by using the masked browsing matrix. As shown in Table 5, when they previewed whole pages, there were two patterns: "browse the whole blocks (pattern 1)" and "browse blocks from the former to middle parts in the whole pages (pattern 2)".

In the pattern 1, we guess that students were very diligent because they clicked whole blocks to preview the contents of the digital textbook. Consequently, they got the highest exercise and report scores. On the other hand, students in the pattern 2 were not able to preview whole blocks in the pages well, especially, page 8 to 19. The contents of these pages include how to use memory allocation and freeing. This is an inevitable in programming a list structure. Therefore, the understanding level of the students in the pattern 2 were poorer than pattern 1 as shown in table 6 because they really did not browse whole blocks in the pages.

In summary, this study newly found two browsing patterns as shown in table 7. In considering students' learning behaviors, we believe that these findings can improve their learning behaviors with teacher support.

Table 7. The summarized browsing patterns

	Browsing matrix	Masked browsing matrix
Browse the whole pages	☑	☑
Browse the pages from 1 to 7	☑	☑
Browse the pages from 1 to 19	☑	☑
Browse the pages from 1 to 30	☑	☑
Browse the pages from 1 to 36	☑	☑
Browse the whole blocks in the whole pages		☑
Browse blocks from the former to middle parts in the whole pages		☑

5. Conclusion and future work

This papers newly proposed a data collection regarding digital textbooks. By using our digital textbook system, we can collect logs of whether students browsed the contents in a page in digital textbooks. This study conducted NMF to find newly several students' learning behaviors compared with learning behaviors found from previous digital textbook systems. We found out that NMF could provide reasonable decomposed matrices to explain browsing patterns.

As a result, we found two learning patterns: "browse the whole blocks (pattern 1)" and "browse blocks from the former to middle parts in the whole pages (pattern 2)". However, it is yet to be conducted the evaluation experiment using the findings. In addition, we need to investigate the appropriate number of patterns because we predefined the number of patterns in this paper.

6. ACKNOWLEDGMENTS

Our thanks to ACM SIGCHI for allowing us to modify templates they had developed.

7. REFERENCES

[1] Fang, H., Liu, P. and Huang, R. 2011. The Research on E-book-oriented Mobile Learning System Environment Application and Its tendency, *International Conference on Computer Science and Education*, 1333- 1338.

[2] Shin, J.A. 2012. Analysis on the digital textbook's different effectiveness by characteristics of learner", *International Journal of Education and Learning*, Vol.1, No.2, 23-38

[3] Kiyota, M., Mouri, K., Uosaki, N. and Ogata, H. 2016. AETEL: Supporting Seamless Learning and Learning Log Recording with e-Book System, Proc. Of the 24th *International Conference on Computers in Education*, 380-385.

[4] MEXT, Japanese Ministry of Education, Culture, Sports, Science and Technology. 2012. *The Vision for ICT in Education*, http://www.mext.go.jp/b_menu/houdou/23/04/__icsFiles/afieldfile/2012/08/03/1305484_14_1.pdf.

[5] Mouri, K., Yin, C. 2017. E-book-based learning analytics for improving learning materials, *IIAI International Congress on Advanced Applied Informatics*, 9-13.

[6] Shimada, A., Mouri, K. and Ogata, H. 2017. Real-time Learning Analytics of e-Book Operation Logs for On-site Lecture Support, *International Conference on Advanced Learning Technologies*, 274-275.

[7] Mouri K., Okubo, F., Shimada, A. and Ogata, H. (2016b). Bayesian Network for Predicting Students' Final Grade Using e-Book Logs in University Education, *International Conference on Advanced Learning Technologies*, pp.85-89.

[8] Ogata, H., Oi, M., Mouri, K., Okubo, F., Shimada, A., Yamada, M., Wang, J. and Hirokawa, S. (2017). Learning Analytics for E-Book-Based Educational Big Data in Higher Education, *Smart Sensors at the IoT Frontier*, 327-350.

[9] Kiyota, M., Mouri, K. and Ogata, H. 2015. Proposal of e-Book Based Seamless Learning System, *Workshop of the 23nd International Conference on Computers in Education*, 611-616.

[10] Mouri, K., Ogata, H. and Uosaki, N. 2017. Learning analytics in a seamless learning environment, *International Learning Analytics & Knowledge Conference*, 348-357.

[11] Desmarais, M. 2011. Conditions for effectively deriving a q-matrix from data with non-negative matrix factorization, *International Conference on Educational Data Mining*, 41-50.

[12] Shimada, A., Okubo, F. and Ogata, H. (2016). Browsing-pattern mining from e-book logs with non-negative matrix factorization, international Conference on Educational Data mining, 636-637.

Hyperparameter Optimization of Machine Learning Models for Educational Datasets

Amritanshu Agrawal, Yiqiao Xu, Abhinav Nilesh Medhekar, Collin F. Lynch
Computer Science Department
NCSU, Raleigh, NC, USA
{aagrawa8, yxu35, amedhek, cflynch}@ncsu.edu

ABSTRACT

Trained models such as Support Vector Machines are governed by parameters. Appropriate parameter settings can make the difference between success and failure. Identifying useful parameters is time-consuming and error prone, nor are there always good parameters to find. We evaluated whether or not hyperparameter tuning, via Differential Evolution (DE) provides significant improvements on 4 EDM tasks with 3 data algorithms. DE found near optimal parameter settings for SVMs but not for DT or RF. And there were significant differences in the weights of those top features as observed, ultimate DE may be suitable for other general tuning tasks.

Keywords

Hyperparameter optimization, Differential evolution, Feature importance

1. INTRODUCTION

The availability of student data from learning management systems, course assessments, and online actions, allows us to perform deep analyses of pedagogical strategies, student interventions, and other educational activities. Prior researchers [23, 17] have surveyed the literature and found that the most common tasks in the educational domain have been resolved through data mining techniques. However, if the prediction model cannot reach a high level of performance [10], the results are of no use to anyone. These models come with different parameter settings which, if set correctly, would improve the performance of the model [13, 9]. These improvements may sometimes make the difference between finding a well-structured model, or finding static. Finding good parameters however, it is costly and time-consuming to identify good parameters. Many researchers often rely on default parameter settings which are often package-specific.

We used Differential Evolution(DE) to tune model parameters faster to obtain a near optimal configuration. [24]. We focus on three of the most prevalent models in EDM: decision trees (DT), random forests (RF), and support vector machines (SVM) [20, 23, 17]. We answer two research questions: **RQ1**: **Does tuning improve the performance scores of different models? RQ2**: **Does tuning make features stand out from the rest?**

2. DATA

This study is conducted on data from UCI machine learning repository and kaggle. Table 1 shows the overview of the datasets. The reproduction package of the code and the prepossessed data is available to download from `https://github.com/amritbhanu/EDM591_Hyperparameter`

Table 1: Dataset Overview

Dataset	# of Instances	# of Attributes	Associated Tasks	Area
D1_math	395	33	Regression	Social
D1_portuguese	649	33	Regression	Social
D2	814	102	Classification	Computer
D3	480	16	Classification	Education

The D1 [6][1] is student performance data from UCI which is a regression task to predict final grades. D2 [21][2] covers software engineering teamwork assessment in education setting from UCI. This dataset consists student teamwork data from San Francisco State University. Features used here capture aspects of students performance in a team. And D3 [2, 1][3] was collected by Kaggle which is a platform for predictive modelling and analytic competitions to produce the best models for predicting data.

We preprocessed the datasets as follows: 1) Filled out missing values with their corresponding median values; 2) Converted categorical attributes via One-hot encoding [5]; and 3) Normalized each feature with min-max normalization [18]. For the different datasets, performance measures, and models, we conducted a 5*5 stratified [22, 14] cross-validation study to make our results more robust. And we checked the amount of variance for each models. For implementations of these models, we used the Scikit-Learn toolkit [19] and we relied upon their default parameters as our baseline.

[1] `http://archive.ics.uci.edu/ml/datasets/student+performance`
[2] `http://tiny.cc/dataset_2`
[3] `https://www.kaggle.com/aljarah/xAPI-Edu-Data`

3. METHODOLOGY

3.1 Hyperparameter Optimization

Hyperparameter optimization is the systematic search for ideal parameters for a given model and dataset or learning task. Prior literature suggests some of the most popular optimizers like simulated annealing (SA [8]), various genetic algorithms [12], DE [24], tabu search and scatter search [11, 3, 15], particle swarm optimization [16], etc. make use of Hyperparameter Optimization. This tells that hyperparameter optimization has been vastly studied in the literature and its impact is well understood [4]. Yet issues of tuning are rarely addressed when it comes to EDM domain.

DE is a stochastic based parameter optimization algorithm [24, 7]. DE is used to optimize either the MSE or F1-score. DE operates through similar computational steps as employed by a standard evolutionary algorithm (EA) but there is a key difference on how it evolves its candidates. DE is smarter than other optimization techniques like certain classes of GAs, SA as they mutate all their attributes independently, whereas DE supports vector-level mutation that retain the association between variables in the space [7].

DE generates new Candidates by extrapolating between current solutions from the Population^5. Three solutions a, b and c are selected at random^{12}. For each parameter i, at some probability cr^{13}, we replace the old setting *candidate* with $newf$. We use an extrapolation equation^{17}, $newf = a[i] + f * (b[i] - c[i])$. There is a trim function that limits the new value to the legal range $[min, max]$ of that parameter. With every generation of DE, the Population contains examples that are better than at least one other candidate. As the looping progresses, the Population is full of increasingly more valuable solutions which, in turn, also improves the candidates, which are extrapolated from the Population. Finally, we get the best (near optimal) set of parameters from the DE^9.

3.2 Machine Learning Algorithms

In this study, we analyzed the effect of different parameter sets on three classic machine learning models: *Support Vector Machine*, *Classification and Regression Decision Tree(CART)*, and *Random Forest Tree*. Table 2 shows the parameters and the range we tuned in this study. To evaluate our model output, we apply weighted precision and recall scores for classification tasks. The traditional precision score is the positive prediction value while recall score is the true positive rate. However, in multiple classification tasks, the unbalanced distribution of classes influences the precision and recall values a lot. In certain situations, the minority class result does not have much of an impact on the total scores. To solve this problem and give an advantage to small classes, we apply weighted scores [25] which weights the average of each class by their level of support. In addition, for regression tasks, we apply mean squared error which measures the squared error between the predicted and actual values.

Table 2: Machine Learning Models Parameters

Parameters	Default	Tuning Range
SVM		
C	1.0	[0.1, 100]
kernel	rbf	{linear, poly, rbf, sigmoid}
degree	3	[1, 20]
CART		
min_impurity_decrease	0.0	[0, 1]
min_samples_split	2	[2, 10]
min_samples_leaf	1	[1, 50]
max_depth	None	[1, 20]
Random Forest		
min_impurity_decrease	0.0	[0, 1]
min_samples_split	2	[2, 20]
min_samples_leaf	1	[1, 20]
max_leaf_nodes	None	[2, 50]
n_estimators	10	[10, 150]

Table 3: Model performances for untuned and tuned

MSE	math			portuguese		
	SVM	RF	DT	SVM	RF	DT
D1_Untuned	9.3	1.8	2.9	4.8	1.3	2.3
D1_Tuned	**1.9**	1.9	**2.0**	**1.2**	1.3	1.2
F1 score	SVM		RF		DT	
	P	R	P	R	P	R
D2_Untuned	0.89	0.53	0.91	0.89	0.90	0.90
D2_Tuned	0.91	**0.89**	0.90	0.87	0.91	0.90
D3_Untuned	0.62	0.58	0.78	0.78	0.70	0.70
D3_Tuned	**0.71**	**0.72**	0.79	0.76	**0.75**	0.73

4. RESULTS

All these results were computed on High Performance Computing (HPC) Servers available at NC State. They were run on 16 cores with minimum of 2GB of RAM per core. And we applied 10-fold-cross validation for each experiment to make our study robust.

As mentioned above, D2 and D3 are classification tasks, so we report precision and recall. We report MSE for D1_math and D1_portuguese since they are regression tasks. Table 3 shows the performance comparison of tuned and untuned results for all datasets. We only reported median of 25 repeats for each measure, and the values which are in bold shows statistically significance than others.

Per the D1 results, we found that tuning improves the performance of SVM(with 400% improvement) and DT significantly, however RF do not show any difference. In D2, we observed median precision and recall values indicate tuning improves recall with SVM significantly. Meanwhile, RF and DT do not show much difference. In addition, D3 results shows performance for D3 tuning improves with SVM significantly, while RF and DT do not show much difference for either precision or recall. *Therefore, after examining the results we can say that DE or hyperparameter optimization improves the performance of SVM significantly by finding optimal parameter settings.*

We also note what DE choose as the optimal configurations in those 25 repeats. As previously mentioned, we performed 5*5 cross validation, in which every time DE was run. This way DE finds 25 different parameter settings to find the optimal settings and that is why we report boxplots to show the median and variance of each parameter for DT for the 3 different datasets in figure 1 and for SVM, RF, please see online[4] due to space restrictions.

[4] http://tiny.cc/rq1_parameters

When considering the figures described above, we can see that they find valid settings which fall far off from the default settings of every learner in every dataset. For example, if we look at the parameter min_samples_leaf for DT in Figure 1a, we observe that median value is close to 25, and on the other hand the default value is mentioned to be 1. We observed similar kind of examples for every learner and every other dataset. *This suggests that everytime you use a learner for a new dataset we need to run an optimizer like DE to find the optimal settings rather than using defaults.*

One important consideration for any researcher with hyperparameter optimization is its computational cost. We acknowledge that any optimization study would be expensive but is that justifiable to use. We observed that 1 run of DE usually terminates DT, RF and RF within 1.5, 26.13 and 25.3 seconds respectively for each of the datasets under the study. When we consider SVM, we observed that it improves about 400% for regression analysis at a cost of 25 more seconds. Similar instances are observed for D2, and D3. *Thus it is justifiable to use an optimizer (like DE).*

With respect to the second research question, each model assigns a weight to each feature while modeling the data. Figure 2 shows the top 10 feature importance values comparing untuned and tuned settings for RF for D3. Y-axis shows the name of features and x-axis shows importance of each of these attributes. Due to space restrictions, the results for the other models and datasets can be seen online[5].

After tuning, DT finds few features which stand out completely from the others. Also, with RF it is observe that relative feature importance changes. This kind of trend is also observed for the math and portuguese and for D2.

This concludes that researchers, industrial people and education systems should be wary of using top features find by default models to drive their analysis or product. They might want to use tuning to identify these top features and then use it for their analysis. Thus, we conclude that tuning do impact on the importance of features in educational machine learning models.

5. CONCLUSIONS & FUTURE WORK

Our goal in this paper was to address two research questions. Based upon our results above we conclude that with respect to (RQ1). We observed that hyperparameter tuning did lead to statistically significant differences between SVM and the other models but not between DT and RF. Thus the use of an optimizer like DF is justified given its cost but is not guaranteed to yield major gains. With respect to (RQ2) we conclude that the observed feature importance does change after using tuning and some of the features do stand out from the rest when compared against learner's default settings. This indicates that the use of tuning may be informative on a per-feature basis.

Based on our findings above, we offer some general recommendations. Hyperparameter optimization improves the performance of the models by finding near optimal parameter settings. Default machine learning algorithms settings are incorrect and we should use an optimizer like DE. Different datasets need different configurations to model them accurately and hyperparameter optimization needs to be run everytime if you are using a different data. It cannot replace the default configurations of models from the findings of any other tuning study. Tuning is computationally costly but the performance gain achieved makes this cost an acceptable increase. We can also rightly claim that in future any such studies should involve hyperparameter optimization and DE is a good candidate to find a near optimal solution. As a conclusion, when educational researchers, developers, and instructors make decisions or provide suggestion to students based on any kind of machine learning models, such as feature selection, classification, or regression, they should keep in mind that their machine learning models may not be as reliable as they expect without hyperparameter tuning.

The limitation of this study is the tuning parameters and tuning ranges that are selected by other researchers, so there is still a small chance we achieved the local optima instead of global optima or we failed to choose the most effective parameter settings. In addition, we only analyzed 3 machine learning methods, and among them, decision tree and random forest have the same core algorithm. So in the future, we plan to analyze more machine learning methods, e.g. Naive Bayes, with general benchmark datasets in the educational domain.

[5] `http://tiny.cc/rq2_features`

6. REFERENCES

[1] E. A. Amrieh, T. Hamtini, and I. Aljarah. Preprocessing and analyzing educational data set using x-api for improving student's performance. In *Applied Electrical Engineering and Computing Technologies (AEECT), 2015 IEEE Jordan Conference on*, pages 1–5. IEEE, 2015.

[2] E. A. Amrieh, T. Hamtini, and I. Aljarah. Mining educational data to predict studentâĂŹs academic performance using ensemble methods. *International Journal of Database Theory and Application*, 9(8):119–136, 2016.

[3] R. P. Beausoleil. âĂIJMOSSâĂİ multiobjective scatter search applied to non-linear multiple criteria optimization. *European Journal of Operational Research*, 169(2):426–449, 2006.

[4] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *JMLR*, 13(Feb):281–305, 2012.

[5] J. Brownlee. `https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/`, 2017. [Online; accessed 5-March-2018].

[6] P. Cortez and A. M. G. Silva. Using data mining to predict secondary school student performance. 2008.

[7] S. Das and P. N. Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE transactions on evolutionary computation*, 15(1):4–31, 2011.

[8] M. S. Feather and T. Menzies. Converging on the optimal attainment of requirements. In *Requirements Engineering, 2002. Proceedings. IEEE Joint International Conference on*, pages 263–270. IEEE, 2002.

[9] W. Fu, T. Menzies, and X. Shen. Tuning for software analytics: Is it really necessary? *Information and*

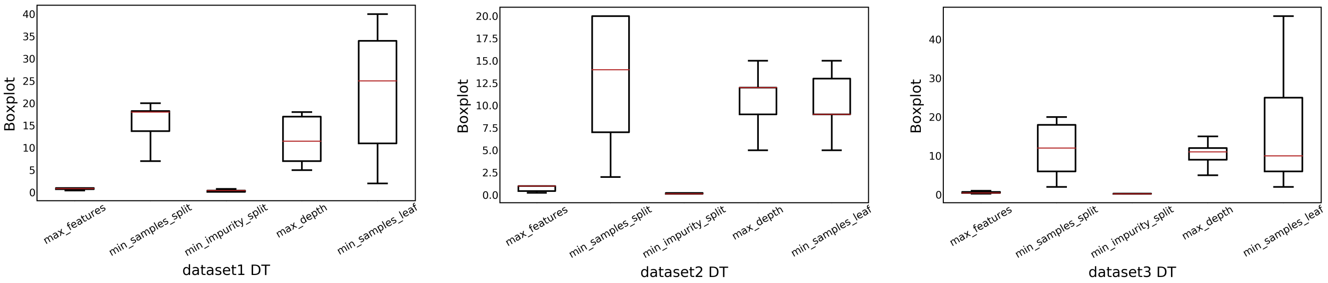


Figure 1a: Tuned values of Decision Tree for D1_portuguese.

Figure 1b: Tuned values of Decision Tree for D2.

Figure 1c: Tuned values of Decision Tree for D3.

Figure 1: Boxplots showing median value (red line) and its variance for each parameter of Decision Tree against 3 datasets. These are calculated for 5*5 cross-validations.

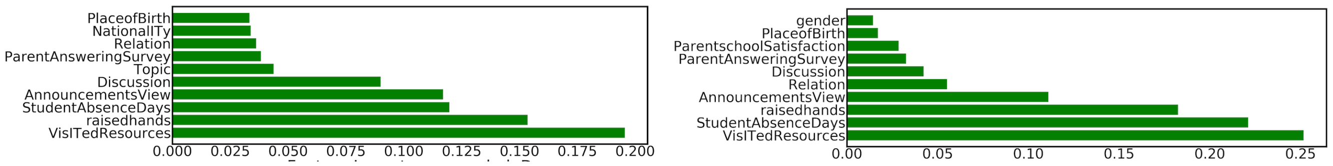


Figure 2a: Untuned.

Figure 2b: Tuned.

Figure 2: Top 10 features importance for D3 for RF

Software Technology, 76:135–146, 2016.

[10] P. Gleason and M. Dynarski. Do we know whom to serve? issues in using risk factors to identify dropouts. *Journal of Education for Students Students Placed At Risk*, 7(1):25–41, 2002.

[11] F. Glover and C. McMillan. The general employee scheduling problem. an integration of ms and ai. *Computers & operations research*, 13(5):563–573, 1986.

[12] A. T. Goldberg. *On the complexity of the satisfiability problem*. PhD thesis, New York University, 1979.

[13] D. F. Gordon and M. Desjardins. Evaluation and selection of biases in machine learning. *Machine learning*, 20(1):5–22, 1995.

[14] R. Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Stanford, CA, 1995.

[15] J. Molina, M. Laguna, R. Martí, and R. Caballero. Sspmo: A scatter tabu search procedure for non-linear multiobjective optimization. *INFORMS Journal on Computing*, 19(1):91–100, 2007.

[16] H. Pan, M. Zheng, and X. Han. Particle swarm-simulated annealing fusion algorithm and its application in function optimization. In *Computer Science and Software Engineering, 2008 International Conference on*, volume 1, pages 78–81. IEEE, 2008.

[17] Z. Papamitsiou and A. A. Economides. Learning analytics and educational data mining in practice: A systematic literature review of empirical evidence. *Journal of Educational Technology & Society*, 17(4):49, 2014.

[18] S. Patro and K. K. Sahu. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*, 2015.

[19] 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. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[20] A. Peña-Ayala. Educational data mining: A survey and a data mining-based analysis of recent works. *Expert systems with applications*, 41(4):1432–1462, 2014.

[21] D. Petkovic, M. Sosnick-Pérez, K. Okada, R. Todtenhoefer, S. Huang, N. Miglani, and A. Vigil. Using the random forest classifier to assess and predict student learning of software engineering teamwork. In *Frontiers in Education Conference (FIE), 2016 IEEE*, pages 1–7. IEEE, 2016.

[22] P. Refaeilzadeh, L. Tang, and H. Liu. Cross-validation. In *Encyclopedia of database systems*, pages 532–538. Springer, 2009.

[23] R. B. Sachin and M. S. Vijay. A survey and future vision of data mining in educational field. In *Advanced Computing & Communication Technologies (ACCT), 2012 Second International Conference on*, pages 96–100. IEEE, 2012.

[24] R. Storn and K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[25] G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer, 2009.

Retrieving IRT parameters with half information

Marie Sacksick[*]
KDIS - University of Cordoba
CHArt - University Paris 8
marie.sacksick@domoscio.com

Sebastián Ventura
KDIS - University of Cordoba
sventura@uco.es

ABSTRACT

Item Response Theory can be used to estimate the degree of mastery of a concept by learners, to automatically assess their knowledge. The models stemming from this theory are tuned to be adapted to the questions used to assess mastery. The correct estimation of the parameters is key to be able to have a correct estimation of the mastery. However, this estimation can be skewed by missing data, noise on the model, or a lack of data.

The question we ask here in this paper is how much data, created by a given number of students answering to a given number of questions is necessary to retrieve reliable coefficients of the questions, when the database at disposal have missing data. To do so we use simulated data. There are two case studies with different levels of data emptiness: one is the baseline and has complete information, the other has only half information.

We find that even though IRT models seem robust against missing values, it is not possible to use the thresholds of the literature obtained with a full database.

Keywords

item response theory, parameter estimation, missing values

1. INTRODUCTION

Item Response Theory (IRT) models are used in psychometrics to evaluate the value of a "latent trait", the value of a descriptor that cannot be assessed directly. IRT offers a framework to be able to measure this unreachable feature. These models are widely used in education to evaluate the degree of understanding and of mastery of a piece of knowledge. In the educational context, that latent trait is called "ability".

As the "ability" cannot be assessed directly, it is necessary to know from which amount of data it is possible for the model to give good estimations. Some studies have been conducted, such as (Chuah, Drasgow, & Luecht, 2006) and (Şahin & Anıl, 2016), to highlight a threshold of data amount under which the results cannot be seen as reliable. To our knowledge, no study has been conducted with a database where the students only answered some of the questions, and not all the database. This situation is very likely to happen, for example when the learner did not have enough time to answer all the questions.

This study uses simulated data, which therefore respects exactly the IRT model. We are investigating whether the IRT algorithm is able to retrieve the simulated questions coefficients. Data is simulated, and cleaned; we run an IRT algorithm thanks to the software R; and finally the theoretical and experimental parameters are compared and the quality of the estimation is estimated through various indicators.

2. RELATED WORK

2.1 What is IRT?

The IRT builds a probabilistic model which hypotheses a relationship between characteristics of the questions and the mastery of the topic by the student. This model has two sets of parameters: the latent trait dedicated to the representation of the student, and some dedicated to the representation of the questions. In this study, we only focus on unidimensional models, and the latent trait can be represented by a unique parameter, usually noted θ.

Here, given a student S_j and given a question Q_i and working with the unidimensional 2-Parameter Logistic model, the probability of success of the student for that question can be written:

$$P(S_j, Q_i) = \frac{e^{(a_i(\theta_j - b_i))}}{1 + e^{(a_i(\theta_j - b_i))}} \tag{1}$$

With $[a_i, b_i]$ being the parameters of the question Q_i. a_i is called the discrimination and it is positive. b_i is called the difficulty; it can either be positive or negative, and 0 represents the mean difficulty.

The inputs of the item response theory models are the answers of the students to the questions. The likelihood of the responses patterns given the probability of success explained in eq. (1) is maximized so as to deduce the most likely questions coefficients and students abilities. Compared to other evaluation theories, one of the advantages of the IRT models is their ability to deal with missing values. There are many methods to estimate the parameters, including Bayesian or non-Bayesian, so we will not list them here.

2.2 Previous work

[*]PhD Student hosted by the society Domoscio

We would like the reader to keep in mind that, depending on the algorithm used to estimate the parameters, the parameters will not be the same, so as their precision; Gao and Chen (2005) give an example of such a situation. Given the evolution in the methods of parameters estimation, we choose not to use studies older than 2000. This drastically reduces the number of studies trying to evaluate the fit of the estimation of the parameters.

RMSD is used in various studies as indicator: (Svetina et al., 2013) and (Yavuz & Hambleton, 2017) use it on simulated data to compare theoretical and experimental values of item parameters; (Svetina et al., 2013) also uses it on person ability. It is also used in (Şahin & Anıl, 2016) on the item parameters where the baseline is the parameters obtained when all the data are used; the estimation using part of the database are said good when $RMSD \leq 0.33$. (Wyse & Babcock, 2016) uses it to have a view on items parameters variations.
Correlation is used as an indicator by (Şahin & Anıl, 2016) like RMSD; the results are said good when $r \geq 0.70$.
Biais can be also used, like in (Svetina et al., 2013).

These studies never paid attention to the influence of missing values, while in education, the databases are continuously filled with missing values. They can have several origins: different sets of questions have been given to the students, the students did not have time to answer the question, they chose to skip it, etc. We choose to focus on this part.

3. EXPERIMENTAL

3.1 Description of experiments

The research question can be formulated as follows: how much data, created by a given number of students answering to a given number of questions is required to obtain a good estimation of the parameters of the questions, given that the students may not answer all the questions of the database? In this study, we do not aim at measuring the goodness of fit of the model, since in the first axis we know that the model is the good one: the data were simulated thanks to it.

3.2 Methods

In that study, we chose to use simulated data. This allows us to know the latent trait of the students and the coefficients of the questions, thus we are able to compare precisely the theoretical coefficients with the experimental ones. Moreover, since the data is simulated thanks to the model which will be applied, there is no interference of model misfit.

Data has been simulated for 50, 100, 500, 1000, 2000, and 3000 students, on 4, 8, 16 and 32 items, which make a total of 24 situations. The abilities of the students follow a standard normal distribution, in this we follow the examples of (Kim, Moses, & Yoo, 2015); (Neel, 2004); (Yavuz & Hambleton, 2017). The discriminations of the items follow a uniform distribution between 0.8 and 1.8, in this we follow the examples of (Svetina et al., 2013); (Yavuz & Hambleton, 2017). The difficulties of the items follow a standard normal distribution, in adequacy with the abilities, in this follow the examples of (Haberman, Sinharay, & Chon, 2013); (Svetina et al., 2013).

For a student S_j and an item Q_i this probability P_{ij} is computed by equation 1 and compared to random number computed following a uniform law between 0 and 1. If it is above, the student answered the question correctly, otherwise it is false.

The parameters have been computed thanks to the package **mirt** in R, with an "itemtype" selected at "2PL" which refers to the 2-Parameters Logistic model.

3.3 Data cleaning

The parameters of a question cannot be evaluated if it has never been answered, or if all the students answered the same thing (i.e. if they all succeeded or they all failed to that question). The data is not simulated to avoid that situation, since it could introduce bias. Instead, we remove the question of the database: in IRT terms, this kind of question is useless because it does not add any information.

3.4 Cases studies

The study has been separated in two cases.

Case A. We have full data, which means that all the students answered all the question, there is no missing value. This case is designed to be the baseline, the "perfect case".

Case B. The students only answered half of the questions. Each student answers a to a different random subset of questions, without checking the number of student who already answered the question, nor the difficulty or discrimination of the questions.

3.5 Indicator

The results are shown in the following figs. 1 to 4. The indicator is the RMSD between the experimental and theoretical values of the questions' coefficients, which one wants as low as possible.
In accordance with the literature, we chose the value 0.3 as the threshold for the RMSD (Şahin & Anıl, 2016). In the following plots, it is represented by a bar.
We represent the results of the difficulty and discrimination parameters for the two cases A and B.

4. RESULTS AND DISCUSSION

4.1 Experiment of axis 1

4.1.1 Results

The results of case A are shown in figs. 1 and 2. The results of case B are shown in figs. 3 and 4.

4.1.2 Discussion

In the two cases, we can see that the difficulty parameters are always easier to compute than the discrimination ones. This is a phenomenon frequently noticed in the literature. As Svetina et al. (2013) points out, the RMSD of the difficulty would have been bigger if we had chosen $b \rightsquigarrow N(0, 2)$ instead of $b \rightsquigarrow N(0, 1)$ because of the imprecisions "in the long tail", i.e. for low or high difficulties.

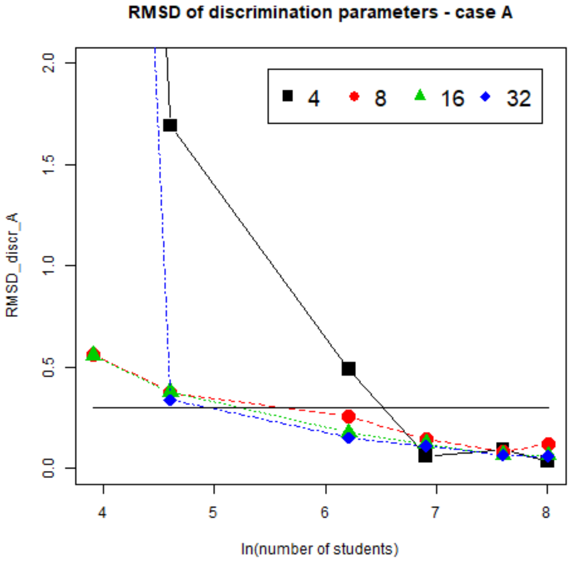


Figure 1: RMSD of discrimination parameter in case A

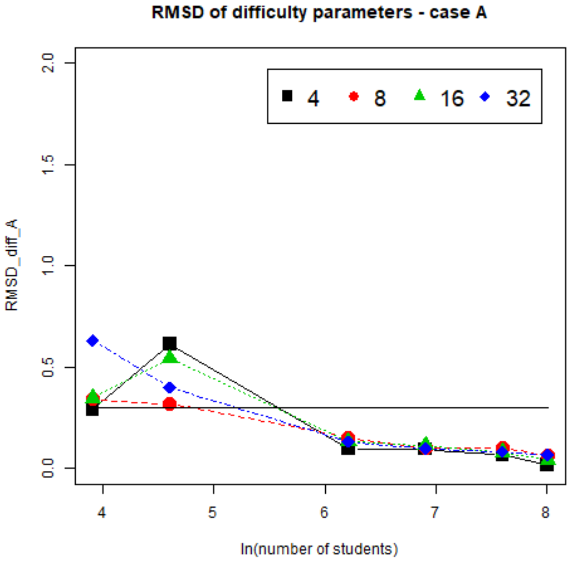


Figure 2: RMSD of difficulty parameter in case A

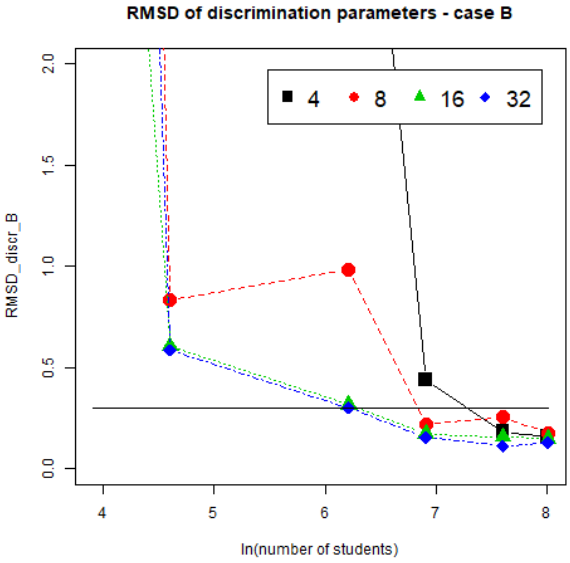


Figure 3: RMSD of discrimination parameter in case B

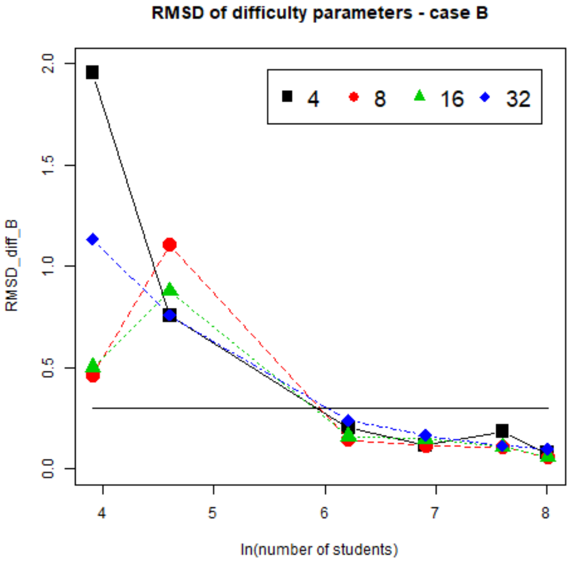


Figure 4: RMSD of difficulty parameter in case B

In case A, we confirm the findings of the literature and extend it to that package, which states that from 500 students and 8 questions the results are good. However with only 4 questions, the results are a little bit weak, as can be seen with the discrimination parameter.

Case B highlights that both the number of students and the number of questions are important parameters. This is less noticeable in case A because it converges towards acceptable situation too quickly. In case B, it can be noticed by looking at the curve representing the RMSD and the correlation of the discrimination parameters.

Case B brings out that the relationship between the percent of answers and the amount of data required might be linear. Here we only have half of the data, and for the same number of student we need twice as much questions to obtain the same quality of results, and the same holds for the situation with the same number of questions, twice as much students are required to obtain the same quality of results.

4.1.3 Conclusion

The main lesson is that when we deal with a database with missing values, we cannot use the thresholds of the literature obtained with a full database.

5. CONCLUSION AND FUTURE WORK

In this study, we aimed at understanding the effects of missing values on the reliability of parameters estimation, and the threshold of data amount. We highlighted that missing data is a parameter that has to be taken into account when one uses a database, and that the thresholds of the literature obtained with a full database cannot be used.

When facing case A, we recommend to have at least 4 questions with 1000 students, or 4 questions with 500 students; when facing case B, we recommend to have at least 8 questions with 1000 students, or 4 questions with 2000 students.

To complete this study, we will go through other cases of missing data, and use other indicators, such as correlation. That study made the hypothesis that the data respect exactly the model: we will investigate the influence of noisy data. One could also compare these results with other programs, whether other libraries in R or software such as WINSTEP or PARSCALE.

6. ACKNOWLEDGMENTS

The authors wish to thanks Charles Tijus and Simon Lemerle for their help.

This study has been partially financed by the National Association of Research and Technology of France, and Domoscio, which we thank too.

This work has been partially supported by the Spanish ministry of Economy and Competitiveness and the European Regional Development Fund, grant TIN 2017-83445-P.

References

Chuah, S. C., Drasgow, F., & Luecht, R. (2006). How big is big enough? Sample size requirements for CAST item parameter estimation. *Applied Measurement in Education*, *19*(3), 241–255. Retrieved from http://www.tandfonline.com/doi/abs/10.1207/s15324818ame1903_5

Gao, F. & Chen, L. (2005). Bayesian or non-Bayesian: A comparison study of item parameter estimation in the three-parameter logistic model. *Applied Measurement in education*, *18*(4), 351–380. Retrieved from http://www.tandfonline.com/doi/abs/10.1207/s15324818ame1804_2

Haberman, S. J., Sinharay, S., & Chon, K. H. (2013). Assessing item fit for unidimensional item response theory models using residuals from estimated item response functions. *Psychometrika*, *78*(3), 417–440. Retrieved from http://link.springer.com/article/10.1007/s11336-012-9305-1

Kim, S., Moses, T., & Yoo, H. H. (2015). A comparison of IRT proficiency estimation methods under adaptive multistage testing. *Journal of Educational Measurement*, *52*(1), 70–79. Retrieved from http://onlinelibrary.wiley.com/doi/10.1111/jedm.12063/full

Neel, J. H. (2004, November). A New Goodness-of-Fit Test for Item Response Theory. *Journal of Modern Applied Statistical Methods*, *3*(2), 581–593. doi:10.22237/jmasm/1099268760

Şahin, A. & Anıl, D. (2016, December). The Effects of Test Length and Sample Size on Item Parameters in Item Response Theory. *Educational Sciences: Theory & Practice*, *17*(1). doi:10.12738/estp.2017.1.0270

Svetina, D., Crawford, A. V., Levy, R., Green, S. B., Scott, L., Thompson, M., . . . Kunze, K. L. (2013). Designing small-scale tests: A simulation study of parameter recovery with the 1-PL. *Psychological Test and Assessment Modeling*, *55*(4), 335. Retrieved from https://pdfs.semanticscholar.org/c2f5/0394c4d75b98d7b7ccab91d9d429c pdf

Wyse, A. E. & Babcock, B. (2016, June). How Does Calibration Timing and Seasonality Affect Item Parameter Estimates? *Educational and Psychological Measurement*, *76*(3), 508–527. doi:10.1177/0013164415588947

Yavuz, G. & Hambleton, R. K. (2017, April). Comparative Analyses of MIRT Models and Software (BMIRT and flexMIRT). *Educational and Psychological Measurement*, *77*(2), 263–274. doi:10.1177/0013164416661220

Semantic Matching Evaluation in ElectronixTutor

Colin Carmon
University of Memphis, Institute for Intelligent Systems
365 Innovation Drive
Memphis, TN 38152
cmcarmon@memphis.edu

Brent Morgan
University of Memphis, Institute for Intelligent Systems
365 Innovation Drive
Memphis, TN 38152
brent.morgan.phd@gmail.com

Andrew J. Hampton
University of Memphis, Institute for Intelligent Systems
365 Innovation Drive
Memphis, TN 38152
jhmpton8@memphis.edu

Zhiqiang Cai
University of Memphis, Institute for Intelligent Systems
365 Innovation Drive
Memphis, TN 38152
zcai@memphis.edu

Arthur C. Graesser
University of Memphis, Institute for Intelligent Systems
365 Innovation Drive
Memphis, TN 38152
graesser@memphis.edu

ABSTRACT

Intelligent tutoring systems (ITS) aim to optimize the learning experience by adapting to the needs of the individual learner. However, the system may not always adapt appropriately. Meta-assessment of learner responses in ITSs can improve instruction efficacy and learner satisfaction. Accordingly, this paper evaluates the quality of semantic matching between learner input and the expected response in AutoTutor, an ITS which holds a conversation with the learner in natural language. AutoTutor's dialogue is driven by the AutoTutor Conversation Engine (ACE), which uses a combination of Latent Semantic Analysis (LSA) and Regular Expressions (RegEx) to assess learner input. We assessed ACE via responses from eighty-six Amazon Mechanical Turk users, who answered 40 electronics questions. This produced a total of 1840 responses, from which we randomly selected 194 responses for our sample. We computed LSA and RegEx scores for learner responses, and two subject-matter experts also judged each response. These analyses explore (1) the relationship between regular expressions and LSA, (2) interrater reliability between the two judges, and (3) the agreement between responses in human judgment and ACE scores computed using regular expressions and LSA. As expected, regular expressions and LSA had a moderate, positive relationship. Also as expected, the agreement between ACE and the human judges was encouraging, but somewhat lower than the agreement between the two humans.

Keywords

AutoTutor, natural language processing, intelligent tutoring systems, meta-assessment, computational linguistics.

1. INTRODUCTION

ITSs that incorporate natural language processing aim to accomplish human-like language processing to properly evaluate user verbal contributions and respond in an appropriate manner. An ideal natural language processing system would be able to paraphrase an input text, translate the text into another language, answer questions about the contents of the text, and draw inferences from the text (Liddy, 2011). ITSs provide individualized instruction and feedback to learners, typically without much variation from human tutoring. Large effect sizes regarding instructional efficacy have been observed in modern ITSs (*d* = .80; VanLehn, 2011). ITSs can cover a wide range of domains, including physics (AutoTutor, Graesser et al., 2004; Nye, Graesser, & Hu, 2014), scientific reasoning (Operation: ARIES, Cai et al., 2011; Operation: ARA, Halpern, Millis, Graesser, Butler, Forsyth, & Cai, 2012), biology (GuruTutor, Olney et al., 2012), and electronics (SHERLOCK, BEETLE-II; Lesgold, Lajoie, Bunzo, & Eggan, 1992; Dzikovska, Steinhauser, Farrow, Moore, & Campbell, 2014). ITSs are often less costly than human tutors in terms of time invested (Dorça, 2015), and, depending on the knowledge domain or task, may combat a shortage of available human tutors. Although ITSs can be costly and time-consuming to develop, one recent approach is to broaden the coverage of topics and implement more learning resources for existing ITSs.

For example, a new ITS, ElectronixTutor (Graesser et al., 2017), integrates multiple ITSs and learning resources to focus on electrical engineering. ElectronixTutor was developed as part of the ONR STEM Grant Challenge. ElectronixTutor is designed for select trainees who scored above average in the Armed Services Vocational Aptitude Battery and are in the process of completing electronics courses in A-school as conducted by the Navy Educational Training Command. ElectronixTutor integrates several learning resources and pedagogical strategies to teach students. Some of these learning resources from various intelligent systems include AutoTutor, Dragoon (VanLehn, Wetzel, Grover, & van de Sande, 2016), Learnform (BBN/Raytheon), ASSISTments (N. Heffernan and C. Heffernan, 2016), and BEETLE-II. Additionally, ElectronixTutor offers topic summaries as well as the Navy Electronics and Electricity Training Series (NEETS) for learners to read. Many of the same AutoTutor materials that were made for use with the ElectronixTutor system have been integrated into the Personal Assistant for Life-Long Learning (PAL3; Swartout et al., 2016). PAL3 is an intelligent learning guide that knows a learner's background, skills, and goals and will accompany them throughout their career. Like ElectronixTutor, PAL3 was developed for technician trainees in the Navy. It can also make suggestions on learning materials and difficulty based on its knowledge of the user.

For AutoTutor to properly respond to users in an intelligent manner, it must evaluate user input effectively. AutoTutor's assessment of student input is based on semantic matching, which compares user responses to one or more expected answers. In this paper, we analyze a sample of responses collected from Amazon Mechanical Turk (AMT) workers and discuss the computational linguistics and information retrieval algorithms used to automatically compute semantic matches in user responses to questions or partial questions. Additionally, we compared the system's evaluations to those of subject matter experts.

Section 2 describes conversations in AutoTutor, and Section 3 explains how the system evaluates the accuracy of the learner's input. In Section 5, we detail the methodology used in our analyses. In Section 6, we report the meta-assessment of the system, and discuss the results in Section 7.

2. AUTOTUTOR CONVERSATIONS IN ELECTRONIXTUTOR AND PAL3

AutoTutor teaches by holding a conversation with the learner in natural language. AutoTutor conversations can be a dialogue between the human learner and a tutor agent, or a peer student agent can be added to create a trialogue. Trialogues offer more flexibility in the conversation, including vicarious learning, competition, and contradiction (Graesser, Cai, Morgan, & Wang, 2017).

In traditional classroom environments, students are often assessed with multiple-choice tests. However, multiple-choice formats are mundane and rarely provide the immediate, individualized feedback that comes with a conversational ITS. In contrast, AutoTutor helps the learner actively construct an answer to the question by collaboratively improving on the answer in a turn-based conversation similar to human tutors (Graesser, D'Mello, Hu, Cai, Olney, & Morgan, 2012).

When asking a question, human tutors often identify expectations (good answers or procedural steps) and misconceptions (common incorrect answers) associated with the question. AutoTutor's Expectation and Misconception Tailored Dialogue models the learner's knowledge by matching the open-ended responses to a pre-defined list of expectations required to answer the main question and associated misconceptions. The following is an example of a main question in ElectronixTutor, the ideal answer, and a breakdown of the ideal answer into expectations:

> ***Main Question:*** *What are the I-V characteristics related to the threshold and breakdown voltage of a real diode compared to an ideal diode?*
> ***Ideal Answer:*** *An ideal diode has a threshold voltage of zero. An ideal diode has no breakdown voltage. A real diode has a threshold voltage greater than zero. A real diode has a breakdown voltage less than zero.*
> ***Expectation One:*** *An ideal diode has a threshold voltage of zero.*
> ***Expectation Two:*** *An ideal diode has no breakdown voltage.*
> ***Expectation Three:*** *A real diode has a threshold voltage greater than zero.*
> ***Expectation Four:*** *A real diode has a breakdown voltage less than zero.*

AutoTutor elicits each expectation from the learner via a series of dialogue movies, including pumps, hints, prompts, assertions, and answering student questions. As the dialogue progresses, the tutor provides more and more information to help the learner until the expectation is covered. Feedback is provided to the learner after each dialogue turn. Once an expectation has been covered, the system moves to another uncovered expectation, or, if all other expectations have been covered, to a summary of the entire answer. Table 1 provides an example of hints and prompts used in ElectronixTutor.

Table 1: Hints and prompts for the expectation "An ideal diode has a threshold voltage of zero."

Question Type	Question	Correct Answer
Hint	Consider the I-V voltage parameters. Why does the ideal diode conduct current immediately after the forward voltage is applied to it?	Because it has a threshold voltage of zero.
Hint	Look at the figure on the left. What specific voltage cut-off point does the origin represent for the forward bias voltage?	The threshold voltage of the ideal diode.
Prompt	An ideal diode starts conducting immediately when the applied forward voltage crosses which zero-valued voltage of the diode?	The threshold.
Prompt	Which diode has a threshold voltage of zero?	The ideal.
Prompt	The threshold voltage for an ideal diode is equal to what?	Zero.

3. LATENT SEMANIC ANALYSIS AND REGULAR EXPRESSIONS

Latent Semantic Analysis (LSA, Landauer et al., 2007) is a mathematical technique with 100 to 500 statistical dimensions for assessing the similarity of pairs of texts expressed in natural language. "Cat" and "dog", for example, often appear in the same documents and, as such, have high semantic similarity. The LSA algorithm measures the similarity between a learner's input and the good answer in the form of a cosine match score from 0 to 1.

In addition to LSA, AutoTutor's learner input evaluations also employ regular expressions (Jurafsky & Martin, 2008). Regular expressions are text strings which define a complex search pattern. These strings allow for increased flexibility in recognizing student input in three ways. First, they can account for common misspellings (e.g., "sou?r[cs]\w*" would capture "source", "sourse" "sorce", etc). Second, regular expressions can account for anticipated synonyms (e.g., "increased", "higher", "larger", etc.), Third, they also can handle complex student answers. For example, "A will increase, and B will decrease" can be expressed by the combination of "A.*B, increase.*decrease" and "B.*A, decrease.*increase". This also captures "B will decrease and A will increase", but does not capture "A will decrease and B will increase." Thus, whereas regular expressions capture keywords, synonyms, and complex structures, LSA compares the semantic similarity of the learner's answer to the good answer.

4. THE CURRENT STUDY

This paper evaluates the quality of the semantic matching between learner input and the expected response in AutoTutor. Regular expressions and LSA combined provide the tools required for the analysis, yielding precision and recall scores to compare the models. We expected the ranges of agreement would conform to similar studies in different domains (e.g., Gautam, Swiecki, Shaffer, Graesser, & Rus, 2017), where precision reached 96%, and recall 78%. We analyzed a corpus of responses obtained on AMT to explore (1) the relationship between regular expressions and LSA, (2) interrater reliability between the two human judges, and, most importantly, (3) the agreement between the human judges and scores from ACE computed using RegEx and LSA. We first hypothesized that RegEx and LSA should yield a moderate, positive relationship. Although regular expressions and LSA both compare learners' answers to expectations, they use different approaches. Hence, an especially strong relationship would indicate that using both methods would be redundant. Secondly, we hypothesized that analyses should yield a relatively high agreement for interrater reliability between the human judges. The judges were both subject-matter experts and were expected to reliably distinguish between correct, partial, and incorrect answers. Finally, we hypothesized ratings between ACE analysis and humans would be similar, but lower than interrater reliability between humans. Although automatic assessment continues to improve, human subject-matter experts are still the gold standard.

5. METHOD

We collected data from 86 unique AMT workers who answered 40 questions asked by AutoTutor in ElectronixTutor. Each question received up to 20 user responses, for a total corpus of 1840 responses. Workers were asked to describe their background in electronics and to answer questions to the best of their ability without doing any research. Users were compensated $1 for each response submitted. Of the 1840 collected responses, 194 were randomly selected, roughly 5 from each of the 40 questions. Two subject-matter experts independently rated the user responses on a continuous scale ranging from 1–6. The scoring definitions are displayed in Table 2.

Table 2: Operational definitions for human judge ratings.

1	No attempt to answer the question.
2	Answer is not on topic/includes metacognitive.
3	Answer is on topic, but completely incorrect.
4	Answer is mostly incorrect.
5	Answer is mostly correct.
6	Answer is completely correct.

6. RESULTS

We began by examining the relationship between RegEx and LSA using a Pearson correlation. We hypothesized a moderate, positive relationship between the two, and this was indeed the case, r (194) = .420, $p < .001$. See Table 3 for descriptive statistics.

Table 3: Descriptive statistics for RegEx and LSA.

	Mean	SD	N
RegEx	.328	.354	194
LSA	.477	.269	194

In evaluating learner responses, successfully detecting partial answers can help the system select the best hint or prompt, but the critical decision is whether the response is fully accurate or not. Accordingly, to compare the human judges to ACE, all ratings were recoded to either a 1 (correct) or a 0 (incorrect). For the human judges, any judgment between 1 and 5 was coded as a 0 and a score of 6 as a 1. For Regex and LSA, a threshold of .8 for either score was scored as a 1, and below .8 was coded as a 0. The .8 threshold is subject to change in future studies, but as observed in our data, setting the threshold too low (>.75) often results in a false-positive in evaluating user verbal responses on the part of LSA.

We next analyzed the ratings of the human judges on the responses from AMT users using Cohen's kappa. The interrater reliability was moderate to good, $k = .699$, $p < .001$, but perhaps slightly lower than expected.

Finally, the third analysis examined the agreement between human judge ratings and ACE scores on correct vs. incorrect responses. The interrater reliability between ACE and the first judge was moderate, $k = .509$, $n = 194$, $p < .001$. The interrater reliability between ACE and the second judge was similar, $k = .477$, $n = 194$, $p < .001$. See table 4 for crosstabulation between the human judges and ACE.

Table 4: Agreement among human judges and ACE.

		Judge2		ACE	
		Correct	**Incorrect**	**Correct**	**Incorrect**
Judge1	**Correct**	30	14	24	20
	Incorrect	5	145	11	139
Judge2	**Correct**	—		20	15
	Incorrect			15	144

7. DISCUSSION

This paper investigated both human and automated assessments of AMT workers' responses to electronics questions. The first analysis examined the relationship between two automated methods, LSA and RegEx, which provide complementary evaluations using different components of the text. Hence, although there was a significant relationship, some variation is expected between regular expressions and LSA.

The interrater reliability between the two human judges was moderate to good. The interrater agreement between ACE and the human judges was somewhat smaller, as expected. However, this is nonetheless encouraging because numerous optimizations can be made to increase the agreement, including adding synonyms to the regular expressions and optimizing the threshold for each answer.

The findings from the analyses can be used to train computational models to evaluate the quality of learner contributions more efficiently. Expectation Misconception Tailored conversations from ElectronixTutor covered in this paper focused on main questions and expectations rather than full trialogues including hints and prompts. Due to the nature of the length of answers, it might be beneficial to consider an analysis between main question and expected responses rather than randomly selecting from a pool of both. Having comparable scores from other learners in ElectronixTutor materials adds a definitive boost in evaluating learners' verbal contributions.

Aside from collecting more data from learners to improve evaluation of responses, the system would also need to be further tested to ensure proper functioning. Materials should be prepared by subject-matter experts and accompanied by regular expressions and LSA when collecting data from learners. Collecting more data samples from learners as well as refining RegEx strings will assist in optimizing assessment models. Although agreement has room for improvement between human judges and ACE, the results are ultimately encouraging and certain to improve moving forward.

ACKNOWLEDGMENTS

This research was supported by the Office of Naval Research (N00014-00-1-0600, N00014-15-P-1184; N00014-12-C-0643; N00014-16-C-3027) and the National Science Foundation Data Infrastructure Building Blocks program (ACI-1443068). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of ONR or NSF.

8. REFERENCES

[1] Cai, Zhiqiang & Graesser, Arthur & Forsyth, Carol & Burkett, Candice & Millis, Keith & Wallace, Patricia & Halpern, Diane & Butler, Heather. (2011). Trialog in ARIES: User Input Assessment in an Intelligent Tutoring System.

[2] Dorca, F. (2015). Implementation and use of simulated students for test and validation of new adaptive educational systems: A practical insight. International Journal of Artificial Intelligence in Education 25, 319-345.

[3] Dzikovska, M., Steinhauser, N., Farrow, E., Moore, J., & Campbell, G. (2014). BEETLE II: deep natural language understanding and automatic feedback generation for intelligent tutoring in basic electricity and electronics. Int J Artif Intell Educ, 24, 284–332.

[4] Gautam, D., Swiecki, Z., Shaffer, D. W., Graesser, A. C., & Rus, V. (2017). Modeling classifiers for virtual internships without participant data. In X. Hu, T. Barnes, A. Hershkovitz, L. Paquette (Eds), Proceedings of the 10th International Conference on Educational Data Mining (pp.278-283). Wuhan, China: EDM Society.

[5] Graesser, A.C., Cai, Z., Morgan, B., Wang, L. (2017). Assessment with computer agents that engage in conversational dialogues and trialogues with learners. *Computers in Human Behavior*.

[6] Graesser, A. C., D'Mello, S., Hu, X., Cai, Z., Olney, A., & Morgan, B. (2012). AutoTutor. In *Applied Natural Language Processing: Identification, Investigation and Resolution* (pp. 169-187). IGI Global.

[7] Graesser, A.C., Hu, X., Nye, B., VanLehn K., Kumar, R., Heffernan, C., Heffernan, N., Woolf, B., Olney, A.M., Rus, V., Andrasik, F., Pavlik, P., Cai, Z., Wetzel, J., Morgan, B., Hampton, A.J., Lippert, A.M., Wang, L., Chen, Q., Vinson IV, J.E., Kelly, C.N., McGlown, C., Majmudar, C.A., Morshed, B., and B aer, W. (2017). ElectronixTutor: an intelligent tutoring system with multiple learning resources for electronics. in International Journal of STEM Education: Innovations and Research. January 2017. DOI 10.1186/s40594-017-0072-5.

[8] Halpern, D.F., Millis, K., Graesser, A.C., Butler, H., Forsyth, C., & Cai, Z. (2012). Operation ARA: A computerized learning game that teaches critical thinking and scientific reasoning. Thinking Skills and Creativity, 7, 93-100.

[9] 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 Artif Intell Educ, 24, 470–497.

[10] Jackson, G. T., Ventura, M. J., Chewle, P., Graesser, A. C., and the Tutoring Research Group. (2004). The Impact of Why/AutoTutor on learning and retention of conceptual physics. In J. C. Lester, R. M. Vicari, & F. Paraguacu (Eds.), Intelligent Tutoring Systems 2004 (pp. 501-510). Berlin, Germany: Springer.

[11] Jurafsky, D., & Martin, J. (2008). Speech and language processing. Englewood: Prentice Hall.

[12] Landauer, T. K., McNamara, D. S., Dennis, S., & Kintsch, W. (Eds.). (2007). Handbook of latent semantic analysis. Mahwah, NJ: Erlbaum.

[13] Lesgold, A., Lajoie, S. P., Bunzo, M., & Eggan, G. (1992). SHERLOCK: a coached practice environment for an electronics trouble-shooting job. In J. H. Larkin & R. W. Chabay (Eds.), Computer assisted instruction and intelligent tutoring systems: Shared goals and complementary approaches (pp. 201–238). Hillsdale, NJ: Erlbaum.

[14] Liddy, E.D. 2001. Natural Language Processing. In Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc.

[15] Nye, B. D., Graesser, A. C., & Hu, X. (2014). AutoTutor and family: a review of 17 years of natural language tutoring. Int J Artif Intell Educ, 24(4), 427–469.

[16] Olney, A., D'Mello, S. K., Person, N., Cade, W., Hays, P., Williams, C., Lehman, B., & Graesser, A. C. (2012). Guru: a computer tutor that models expert human tutors. In S. Cerri, W. Clancey, G. Papadourakis, & K. Panourgia (Eds.), Proceedings of Intelligent Tutoring Systems (ITS) 2012 (pp. 256–261). Berlin: Springer.

[17] Swartout, W., Nye, B. D., Hartholt, A., Reilly, A., Graesser, A. C., VanLehn, K., Wetzel, J., Liewer, M., Morbini, F., Morgan, B., Wang, L., Benn, G., & Rosenberg, M. (2016). Designing a personal assistant for life long learning (PAL3). In Z. Markov & I. Russel (Eds.), Proceedings of the 29th International Florida Artificial Intelligence Research Society Conference (pp. 491–496). Palo Alto: Association for the Advancement of Artificial Intelligence.

Large Scale Search for Optimal Logistic Knowledge Tracing Features

Philip I. Pavlik Jr.
University of Memphis
365 Innovation Drive, Suite 303
Memphis, TN 38152-3115
1-901-678-2326
ppavlik@memphis.edu

Neil Zimmerman
McGraw-Hill Education
281 Summer Street, Floor 7
Boston, MA 02210
1-617-859-0054
neil.zimmerman@
mheducation.com

Mark Riedesel
17 Charmark Circle
Middleboro, MA 02346
1-508-944-7557
mark.riedesel@gmail.com

ABSTRACT

In this paper we introduce a new method for creating models of student learning and performance log data using logistic regression. Many such variants have occurred in the literature, but a large section of the space of possible models has not been explored. We use 12 features, combined pair-wise to create 144 variants, and we test these models with 3 versions of these pairwise combinations, using KC models representing the item, objective and item, or objective, item, and student. Each model is fit with a single difficulty constant or one constant to represent each KC. This procedure was replicated in 2 halves of a dataset from a McGraw Hill online practice app. For these data, the findings confirmed the accuracy of rPFA and PFA-Decay features and revealed that the logit feature [ln(successes/failures)] worked well in many contexts to create accurate models. Other decay/forgetting-based features also performed well. Standard PFA did poorly compared to these newer models and the logit feature.

Keywords

Logistic regression; student modeling, adaptive instruction.

1. INTRODUCTION

Discovery of the most parsimonious models for tracing knowledge using logistic regression variants has previously proceeded in a series of steps that can be traced item response theory (IRT). IRT was developed to understand and evaluate results from academic testing by allowing a modeler to fit parameters to characterize performance on a set of items by a set of students. As a form of logistic regression, IRT predicts 0 or 1 (dichotomous) results like the results of individual items for students. In the case of a 1 item parameter IRT model, this result is predicted as a function of student ability minus item difficulty (x), which is scaled to a probability estimate by looking up the probability from the logistic function cumulative distribution, in which $p=(1/(1+e^{-x}))$. It is from this basis that we build our models.

Within this context a “new model” typically consists of new ways to organize the Q-matrix often combined with new ways to compute the effect of prior experience. We search a relatively large space of such logistic regression models in this paper. As it turns out, there are more than 10 plausible ways to compute prior experiences, and these may be applied to the prior experience for different types of encounters.

For example, we can talk about using the natural log of the count of all prior trials for a KC. This predictor will be insensitive to performance and count earlier practice as more effective than later practice according the natural log function. The coefficient should be positive for such a predictor since we expect learning with experience. Another example would be using an exponentially decayed count of prior failures for the KC. This predictor would be more sensitive to recent failures depending on the decay parameter. Because failures, despite the opportunity for learning indicate lower prior performance capability, the coefficient for this predictor is often negative unless there is very strong effective feedback following failures in the learning system generating the data the model is fit to.

2. DATA FROM STUDYWISE

The data used for this study comes from a new mobile application, StudyWise, from McGraw-Hill Education (MHE), which was designed to allow students in courses which use a selection of MHE SmartBook titles to practice the end-of-the-chapter questions anywhere, anytime. At present, there are ten titles available in the Apple App Store and the Google Play Store, the majority of which are from titles related to Biology and Medicine.

Medical-related titles were emphasized because of the continuing need in these fields for practitioners to memorize a great deal of declarative knowledge. In critical situations, doctors, nurses, EMTs, pharmacists, and physician’s assistants need to have knowledge of anatomy, physiology, pharmacology, and other topics memorized for immediate recall and the licensing requirements in these fields reflect this. Most of the user data (about 80%) comes from one title on Anatomy and Physiology.

There is a separate StudyWise app for each title and within an app, the material is organized by topic (i.e. chapter). Each topic has a number of Learning Objectives (LO). Each LO has anywhere from one to five questions associated with it, with the overall average being two distinct questions per LO. We use objectives to represent the KCs in the work below.

3. ANALYSIS

The model space tested used a 3 level KC hierarchy where performance was predicted as a function of 2 predictive feature terms for each of the 3 levels. The simplest model only used an item level KC that traced prior events for exact repetitions for the question item. The next model included a middle level KC that grouped related items in objectives. The most complex model also included a “student” KC that grouped all the items for that learner. To limit the search space, we choose to use the same predictive expression for the two predictive terms in every level of the hierarchy. Thus, if the model used TOC (total opportunity count) for the first term and LNPERF (ln prior failures) for the second term, these same expressions would be used for all three levels, so if the model included item and objective levels, it would use AFM to compute the first term for both levels and use log (prior failures) to compute the second term for both levels. This is a limitation, since a more complete search might reasonably use different predictive expressions at different levels.

Data for these analyses are drawn from 4,694 individual users who worked on StudyWise questions, at their own pacing and schedule, between its launch in early April 2017 and mid-January 2018. The raw dataset contains over a million rows. Unfortunately, analyzing this data set with per-objective difficulties was prohibitively slow due to the memory requirements: to speed the analysis, we examined only the 20 objectives with the most usage; this created a subset of the data consisting of 41 distinct items and 8 distinct topics. The first split contained 1,465 distinct users and 39,570 question-responses and the second split contained 1,517 users and 40,786 question-responses. As shown in our results, our models were quite stable across folds in both test datasets.

Analysis was done in the R programming language. The logistic models were implemented using the built-in generalized linear model (glm) function. For the models that include an exponential or power decay constant, an iterative process was used: a function was written that, for a given decay rate, re-computed the features, refit the model, and calculated and returned the log-likelihood of the model given the data; the built in quasi-Newton optimizer (nlminb) called this function repeatedly until the exponent was found that produced the likeliest model.

3.1 Assumptions

We assume that the models are well tested by fitting the same coefficient for all KCs for each feature. While we do look at intercepts (constants for prior knowledge) for each objective KC, our method does not individualize any slope (learning rate) parameters for each KC, as is normally done, but uses the same coefficient for all KCs for each feature. This essentially means that there is a single learning rate for each of the two learning effect terms at each level, which we fit as an overall coefficient for each of level of our hierarchy (see below). This assumption was necessary at this point in the research to limit the search space. While we do not model difference in learning rates for KCs, we do capture differences in initial item difficulty/prior knowledge at using topic or objective level intercepts. We also compare models using only a single intercept to test the influence of the initial difficulty on determining the best expressions to use in the model.

For these two terms, we allow combinations of any 2 predictors (i.e. successes, failures, or total trials), with the exception that we don't test combinations which use both success or both failures for both predictors (e.g. using count failures and log count failures) is not tested but using total trials for both is allowed (e.g. using count total and log count total) is tested.

We make other assumptions as well, such as limits on our model non-linear parameters (between 0 and 1), however, prior work has typically used values in this range [1,2].

3.2 Features

3.2.1 Total opportunity count (TOC)

This predictor is a simplified equivalent to the well-known AFM model, which predicts performance as a linear function of the prior total experiences with the KC. This is a simplified AFM model, since we do not test using KC to differentiate learning rate, rather, all KCs have the same opportunity count coefficient.

3.2.2 Log total opportunity count (LNTOC)

This predictor has been sometimes uses in prior work and implies that there will be decreasing marginal returns as opportunities increases, according to a natural log function. The assumption here makes the most sense if the exercises provide limited feedback, which might cause lower learning for later practice. [3]

3.2.3 Power-decay TOC (POWTOC)

This predictor multiplies TOC by the age since the first practice (trace creation) to the power of a decay rate (negative power).

3.2.4 Log POWTOC (LNPOWTOC)

This predictor is the same as POWTOC, except it use the natural log of prior trial count plus one.

3.2.5 Exponential decay (EXPTOC)

This predictor considers the effect of the TOC as a decaying quantity according to an exponential function.

3.2.6 Linear PERF (LINEPERF)

This term is equivalent to the terms in performance factors analysis (PFA). [4-7]

3.2.7 Linear sum performance (LINESUMPERF)

This term uses the success minus failures (as term 1) or failures-success (as term 2) to provide the simple summary of overall performance.

3.2.8 Log PERF (LNPERF)

This expression is simply the log transformed performance factor (successes or failures), corresponding to the hypothesis that there are declining marginal returns according to a natural log function. [8,9]

3.2.9 Proportion PERF (PROPPERF)

This expression uses the prior percent correct or incorrect as the predictor. It performed generally well, but not quite as well as the following 3 mechanisms

3.2.10 Logit PERF (LGTPERF)

This function is the log odds with an additional parameter to characterize trial 1 baseline, where logit= log(0/0) otherwise.

3.2.11 Exponential decay of proportion (EXPPROPPERF)

This expression uses the proportion right or wrong (again depending on whether it is expression 1 or expression 2) and was introduced as part of the rPFA model [1,10,11]. We set the number of ghost attempts at 3 as suggested by Galyardt and Goldin [1].

3.2.12 Exponential decay (EXPPERF)

This expression uses the decayed count of right or wrong (again depending on whether it is expression 1 or expression 2). This method appears to have been first tested by Gong, Beck and Heffernan [2]. This method is also part of rPFA and is used for tracking failures only, whereas rPFA uses EXPPROPPERF to track correctness [1].

4. RESULTS WITH OBJECTIVE KCS

For the Objective KC results, the counts for the mid-level KC were based on the objective in the StudyWise application, and when KC intercepts were used they were also based on the objective assignment in the application. Our result took the form of 2 large tables, each containing the 864 models found, note however, that some of these models were partially redundant due to their reflective nature. Since in some of the more complex models, we were not certain the models were fully reflexive, we included them in the analyses. For example, TOC and LNTOC is functionally identical to the LNTOC and TOC.

We first observed that AUC and R^2 correlated at .982, and therefor decided to use AUC for our primary analyses. Next, we confirmed that the 2 randomly selected folds had a highly correlated rank order to insure our results were stable and therefore valid. The Pearson correlation of the AUC rank for the 2 lists was .994. To get a better perspective on which model pairings were doing best, we also looked at the top 10 models split in Table 1.

5. DISCUSSION

The results show that this method produces models that are both novel and compare well with the best models in the literature, e.g. rPFA [1]. We will discuss these results first by reviewing the 12 features and how they fared overall. Following these discussions of the individual features we discuss paired feature models, focusing on models that parallel or replicate models in the literature. In this section we compare how versions of AFM, PFA, rPFA, and PFA-Decay fared vs the models tried in the search.

One implication of these results is to shift the model building discussion away from specific formalism like PFA and towards an approach which tries to find the most effective features for modeling change in performance at each level of the data. The problem with specific formalisms is their coarse grainsize which makes it difficult if not impossible to see how their composition (the features used in the model) influences their function. Due to the complexity of multiple factors all being important to the model, the model may become a black box that might function well but is difficult to improve or understand. Hopefully, by turning the discussion to towards the individual components of models it will be easier for developers to assemble models to fit the special needs of different applications.

5.1.1 Power-decay TOC (POWTOC)

This predictor was among the best features.

5.1.2 Log POWTOC (LNPOWTOC)

Just as the natural log feature improves on the fit of linear AFM model, the natural log also improves the power-decay feature. It seems the diminishing marginal returns for practice quantity combines well with a decaying trace. In fact, while the margin of difference was not large, this feature was the best feature for the objective KC models.

5.1.3 Exponential decay (EXPTOC)

This forgetting predictor did remarkably well for the objective KC models. This suggests that the power-decay model may be less robust in conditions where KCs are coarser grained. It should be noted that the exponential model measured forgetting across trials, while the power law model was a function of time.

5.1.4 Proportion PERF (PROPPERF)

This expression merely uses the prior percent correct or incorrect as the predictor. It does well, performing quite convincingly the objective KC models, however, all three of the following features can be argued to work better in more situations.

5.1.5 Logit PERF (LGTPERF)

This expression uses the logit (natural log of the success divided by failures; or the reverse as the second term). This function paired very well in both datasets, showing general power when combined with either performance or total count features. There is some indication that logit may pair slightly better LNPOWTOC. Also, in the objective KC models we saw that LGTPERF did quite poorly when used in a 3-level model with a single intercept. Apparently using the objective KC intercepts allowed this mechanism to function maximally.

5.1.6 Exponential decay of proportion (EXPPROPPERF)

This expression uses the proportion right or wrong (again depending on whether it is expression 1 or expression 2). As suggested by the inventors of this mechanism, Galyardt and Goldin, we set the number of ghost attempts at 3 [1]. This feature showed strong predictiveness like the LGTPERF or EXPPERF below. This mechanism did seem a bit stronger more generally accurate than logit, similar to EXPPERF below.

5.1.7 Exponential decay (EXPPERF)

This expression uses the decayed count of right or wrong (again depending on whether it is expression 1 or expression 2). This method appears to have been first tested by Gong, Beck and Heffernan [2] and was still used for failures counts in later work [1] with the EXPPROPPERF. This feature showed strong predictiveness like the logit or simple exponential decay above.

Table 1. Objective KC model fit AUC values, averaged across KC hierarchy used (Fold 1).

Model	3 Level
LNPOWTOC EXPPERF	0.7032
EXPPROPPERF EXPTOC	0.7015
LNPOWTOC EXPPROPPERF	0.7012
POWTOC EXPPERF	0.7003
EXPPROPPERF EXPPERF	0.6996
LNTOC EXPPERF	0.6993
PROPPERF EXPPERF	0.6991
EXPPERF EXPPERF	0.6987
EXPTOC EXPPERF	0.6986
PROPPERF EXPTOC	0.6986

Model	Objective and Item
LGTPERF LNPOWTOC	0.6760
LNPOWTOC LGTPERF	0.6760
LNPOWTOC EXPPERF	0.6760
LNPOWTOC EXPPROPPERF	0.6750
LNPOWTOC LINESUMPERF	0.6749
LINESUMPERF LNPOWTOC	0.6749
LNPOWTOC LNPERF	0.6746
LINESUMPERF POWTOC	0.6736
POWTOC LINESUMPERF	0.6736
POWTOC LINEPERF	0.6732

Model	Item
LNPOWTOC EXPPERF	0.6603
LNPOWTOC LINESUMPERF	0.6601
LINESUMPERF LNPOWTOC	0.6601
LGTPERF LNPOWTOC	0.6570
LNPOWTOC LGTPERF	0.6570
LNPOWTOC EXPPROPPERF	0.6561
LNPOWTOC LNPERF	0.6557
LINESUMPERF POWTOC	0.6553
POWTOC LINESUMPERF	0.6553
POWTOC EXPPERF	0.6550

5.2 Best Feature Combinations

Table 1 helps us understand the best ways to pair 2 of the features we looked at in our study. One result that stands out is that the two term model EXPPROPPERF and EXPPERF, corresponding to the form of rPFA's practice features [1] performs very well for the 3 level objective KC models. While this strength was notable, it was also interesting that EXPPROPPERF paired well with EXPTOC also, indicating that any two decaying terms might be best. However, not to be outdone, EXPPERF with EXPPERF also performed very well, indicating the PFA-Decay model [2] was about as accurate as r-PFA for our data, perhaps suggesting the proportion and ghost attempts complexity of rPFA is not needed for accurate models.

6. CONCLUSIONS

In this paper we have shown how to conduct a search over a large space of models to find better features for logistic regression knowledge tracing. This work has revealed at least two features unique to the literature, LGTPERF and LNPOWTOC which competed well with features in some of the latest models. While logistic regression is not a complex formalism compared to methods such as deep knowledge tracing [12], even in complex forms it is computationally efficient and tractable to use in educational systems [13]. For this reason, these results are likely to be practically important for people considering models of learning performance in trial based educational systems.

This project was conducted as part of the development of the LearnSphere community analytic tool [14]. In this paper we searched a space of 864 models, but, we could easily have wanted to search a space of hundreds or thousands of times larger to get a fuller coverage of the possible space of models. Such a search space would have been intractable, requiring years of computer time with our current methods. An alternative to this is to develop some sort of hill climbing search akin to LFA (Leaning Factors Analysis) [15], but such that it performs a search over a subset of the possible categorical variations of model structure.

7. ACKNOWLEDGMENTS

This work is supported by the National Science Foundation Data Infrastructure Building Blocks program under Grant No. (ACI-1443068). Our thanks to McGraw-Hill Education for allowing us to use the data.

8. REFERENCES

[1] Galyardt, A., Goldin, I., 2015. Move your lamp post: Recent data reflects learner knowledge better than older data, *JEDM-Journal of Educational Data Mining* **7** (2015), 83-108.

[2] Gong, Y., Beck, J.E., Heffernan, N.T., 2011. How to Construct More Accurate Student Models: Comparing and Optimizing Knowledge Tracing and Performance Factor Analysis, *International Journal of Artificial Intelligence in Education* **21** (2011), 27-46.

[3] Yudelson, M., Hosseini, R., Vihavainen, A., Brusilovsky, P. 2014. Investigating automated student modeling in a Java MOOC, In: Stamper, J., Pardos, Z.A., Mavrikis, M., McLaren, B. (eds.): *Proceedings of 7th International Conference on Educational Data Mining* (2014), 261-264.

[4] Pavlik Jr., P.I., Cen, H., Koedinger, K.R. 2009. Performance factors analysis -- A new alternative to knowledge tracing, In: Dimitrova, V., Mizoguchi, R., Boulay, B.d., Graesser, A. (eds.): *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, Brighton, England (2009), 531–538.

[5] Gong, Y., Beck, J., Heffernan, N.T. 2010. Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting Procedures, In: Aleven, V., Kay, J., Mostow, J. (eds.): *Intelligent Tutoring Systems*, Vol. 6094. Springer Berlin / Heidelberg (2010), 35-44.

[6] Pavlik Jr., P.I., Yudelson, M., Koedinger, K.R. 2011. Using contextual factors analysis to explain transfer of least common multiple skills, In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.): *Artificial Intelligence in Education*, Vol. 6738. Springer, Berlin, Germany (2011), 256–263.

[7] Pavlik Jr., P.I., Yudelson, M., Koedinger, K.R., 2015. A Measurement Model of Microgenetic Transfer for Improving Instructional Outcomes, *International Journal of Artificial Intelligence in Education* **25** (2015), 346-379.

[8] Chi, M., Koedinger, K.R., Gordon, G., Jordan, P., VanLehn, K. 2011. Instructional Factors Analysis: A cognitive model for multiple instructional interventions, In: Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero , C., Stamper, J. (eds.): *Proceedings of the 4th International Conference on Educational Data Mining*, Eindhoven, The Netherlands, (2011), 61-70.

[9] Lindsey, R.V., Shroyer, J.D., Pashler, H., Mozer, M.C., 2014. Improving students' long-term knowledge retention through personalized review, *Psychological Science* (2014),

[10] Galyardt, A., Goldin, I. 2014. Recent-performance factors analysis, In: Stamper, J., Pardos, Z.A., Mavrikis, M., McLaren, B. (eds.): *Proceedings of 7th International Conference on Educational Data Mining* (2014), 411-412.

[11] Goldin, I.M., Galyardt, A. 2015. Convergent Validity of a Student Model: Recent-Performance Factors Analysis, In: Santos, O.C., Boticario, J.G., Romero, C., Pechenizkiy, M., Merceron, A., Mitros, P., Luna, J.M., Mihaescu, C., Moreno, P., Hershkovitz, A., Ventura, S., Desmarais, M. (eds.): *Proceedings of 8th International Conference on Educational Data Mining*, Vol. 548-5551 (2015), 548-551.

[12] Piech, C., Spencer, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L., Sohl-Dickstein, J., 2015. Deep Knowledge Tracing, *arXiv preprint arXiv:1506.05908* (2015),

[13] Mooney, S., Sun, K., Bomgardner, E. 2017. Predicting Recall Probability to Adaptively Prioritize Study: *NIPS 2017 Workshop: Teaching Machines, Robots, and Humans*, Long Beach, CA (2017),

[14] Stamper, J., Koedinger, K., Pavlik Jr., P.I., Rose, C., Liu, R., Eagle, M., Yudelson, M., Veeramachaneni, K. 2016. Educational Data Analysis using LearnSphere Workshop, In: Rowe, J., Snow, E. (eds.): *Proceedings of the EDM 2016 Workshops and Tutorials co-located with the 9th International Conference on Educational Data Mining*, Raleigh, NC (2016),

[15] Cen, H., Koedinger, K.R., Junker, B. 2006. Learning Factors Analysis - A general method for cognitive model evaluation and improvement: *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*. Springer Berlin / Heidelberg (2006), 164-175.

Using a Hierarchical Model to Get the Best of Both Worlds: Good Prediction and Good Explanation [*]

Kenneth R. Koedinger
Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA 15206
koedinger@cmu.edu

Lu Sun
Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA 15206
ls1@andrew.cmu.edu

Elizabeth A. McLaughlin
Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA 15206
mimim@cs.cmu.edu

ABSTRACT

Understanding how learning transfers from one task to another is a critical topic in learning science. In this paper, we investigate the impact of the scope and granularity of learning transfer by comparing three models across multiple data sets. Prior work demonstrated the value of component models of learning transfer that group items into knowledge components. Within the component models, that work left open whether difficulty (variation in performance over tasks) is better modeled by knowledge components (the strong model) or by items (the "weak" model). The strong component model is theoretically desirable because it provides a single explanation for both difficulty and transfer. However, we find that the weak component model better predicts student performance across six data sets. While this weak model predicts better, it is hard to interpret because an explanatory parameter that represents latent knowledge difficulty of student performance is absent. To maintain explanatory power without sacrificing prediction, we propose a new alternative that uses a hierarchical mixed effect regression model where item difficulty is pooled within component difficulty. Experimental results, across six data sets, show that the predictions of the hierarchical model are better than the strong model and as good as the weak model, while also producing theoretical useful explanatory parameter values for knowledge components.

Keywords

Transfer, hierarchical mixed effects models, student modeling, knowledge component modeling

1. INTRODUCTION

Transfer of learning, the application of knowledge acquired in one situation to other new, relevant learning situations, is an age-old fundamental problem in human cognition and education [1, 8]. A central question is determining the loci of transfer at a grain size of analysis that is fine grained to make accurate and useful predictions yet broad or simple enough to provide explanatory insight that advances science or application. In modeling transfer, [4] contrast two statistical models of the faculty theory of transfer with two statistical models of an alternative component theory of transfer using multiple datasets. That work provided a convincing case for a component theory of transfer over a faculty theory of transfer, it also raised a new question. To be effective, statistical models of learning transfer control both for general student proficiency and variations in the difficulty of tasks. When contrasting statistical models of the component theory, the results were mixed as to whether task difficulty is better modeled by items or by components.

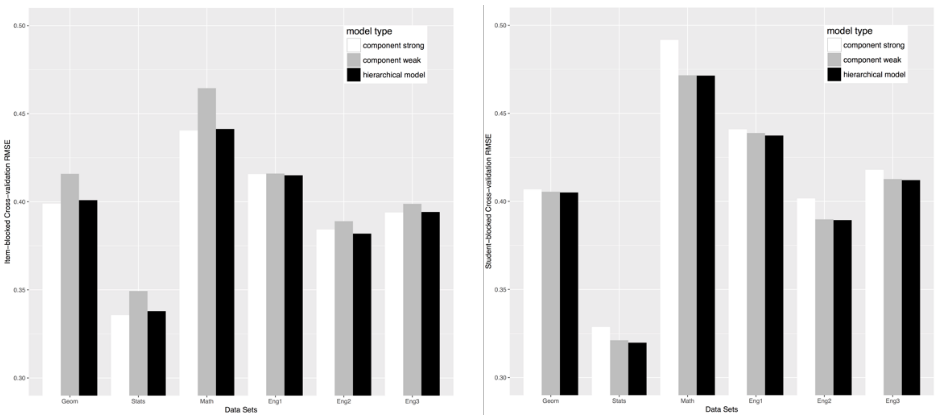
It is worth stating more precise definitions for key terms: item, knowledge component, strong and weak component models. For our purposes, *items* are tasks that appear as questions or steps in problems where student responses are evaluated as correct or not. An example item is to find the area of a circle given its radius is 10. A *knowledge component*(KC) is defined as "an acquired unit of cognitive function or structure that can be inferred from performance on a set of related tasks"[3]. For example, analysis of student correctness data on related geometry tasks leads to inferences about differences in generality of KCs related to trapezoid area versus circle area. Whereas few differences in difficulty across trapezoid items suggests a single general KC: "Use the trapezoid area formula to find any unknown value in the formula", difficulty differences across circle items suggests two KCs that separate items into two groups depending on whether the area is unknown (easier) or the radius is unknown (substantially harder)[5].

A *strong component model*[4] is one in which item difficulty and learning transfer are *both* modeled using KCs. A *weak component model* uses KCs only to model transfer but uses items to model item difficulty. It is "weaker" because it provides a less coherent and less parsimonious theory by having separate explanations for difficulty and transfer rather than a unified explanation as in the strong model. Despite this explanatory disadvantage, the weak model was found to produce better predictions than the strong model in some cases[4]. In particular, it produced better predictions than the strong model when generalizing to unseen students. Given the theoretically desirability of the strong model, we set a goal to develop a statistical model that would maintain the theoretical benefits of the strong model without any sacrifice to prediction. To address this goal, we used hierarchical mixed effects regression as an approach that allows the combination of estimates of both item difficulty and component difficulty.

2. RELATED WORK

Work on statistical student modeling has pursued a variety of alternatives within families of logistic regression variations, of Bayesian Knowledge Tracing variations, and of recurrent neural networks.

[*] (Does NOT produce the permission block, copyright information nor page numbering). For use with ACM_PROC_ARTICLE-SP.CLS. Supported by ACM.



a. Item-blocked cross validation

b. Student-blocked cross validation

Figure 1: Root mean squared error (RMSE) results for item-blocked (a) and student-blocked (b) cross-validation across 6 diverse datasets showing a disadvantage of the weak model in generalization across items (middle gray bars are highest in a) and a disadvantage of the strong model in generalizing across students (left white bars are highest in b). The hierarchical model provides the best prediction fit in most cases (8 of 12) and is essentially tied with the strong model in item generalization in the other cases.

Logistic regression variations include a few simpler alternatives, like Item Response Theory[10] and the Additive Factors Model (AFM). Recently, a family of statistical student modeling based on recurrent neural networks has begun to emerge[7]. Deep Knowledge Tracing is highly complex and is particularly difficult to interpret, thus limiting its explanatory power and application potential, at least at this point. Among these methods, key reasons for contrasting AFM variations are a) to pursue models with a greater bias toward explanatory simplicity as there are many others, as indicated above, pursuing more complex models and better prediction with relatively less concern for interpretability and downstream application and b) to more directly build off the prior work on transfer that used AFM variations but left an open question as described above[4].

3. METHODS

The component theory of transfer uses a matrix to map knowledge components to items. The strong model suggests a single explanation for both difficulty and transfer but sacrifices some predictive power. The weak model theory offers better prediction but is less explanatory. We investigate an alternative model that combines benefits of the strong and weak models in a hierarchical fashion. We hypothesize that this alternative model will provide the explanatory power of the strong model without losing the predictive power sometimes better displayed by the weak model. We seek the interpretability and application benefits of explaining both difficulty and transfer using KCs, but without losing predictive power as has been observed with a strong component model or AFM.

3.1 Datasets

A variety of datasets representing four domains including geometry, algebra, English articles and statistics, with different task ordering approaches, were selected from LearnLab's DataShop [2], an open repository for diverse educational domain data. The datasets had different characteristics (e.g., number of students, number of KCs, etc,), which have been reported previously [4].In the educational technology applications that students used in producing this data, students solve problems or answer questions sometimes with multiple steps with feedback. Each evaluated step is considered a task or assessment "item" and can be labeled with one or more knowledge components (KC). Each dataset had multiple knowledge component models associated with it, where each model represents a different mapping from steps/items to skills/KCs. In this paper, for each dataset we used best KC model generated by LFA. The best was selected using the lowest root mean squared error (RMSE) on item-blocked cross-validation (explained below). We compared three different statistical models across six datasets.

3.2 Metrics of predictive accuracy

To evaluate predictive accuracy, we used five independent runs of 10-fold cross-validation (CV) using three variations of how the folds are produced, randomly, blocked by item, blocked by student, as per standard practice in LearnLab's DataShop[2]. We explain the item-blocked and student-blocked approaches next. The prior work[4] compared the component strong model and component weak model by creating folds in CV and blocking data records either by item or by student. In item-blocked CV, on each iteration, all data for an item is either in the training set or test set, but never both. In that prior work, the prediction fit of the weak models was consistently worse than the strong models when tested for generalization to new items, that is, via item blocked CV. This result can be explained by noting that in the weak model item difficulty estimates are not available for predicting test set data. The weak model relies only on overall difficulty, as well as student proficiency and KC learning rate, to predict on test set. In contrast, the strong model can use KC difficulty, as well as student proficiency and KC learning rate, to predict on test set. At the same time, it is important for models to generalize to new students and thus testing them via student-blocked CV is also sensible. In this case, prior work[4] demonstrated that the strong models were consistently worse than the weak when tested for generalization to new student, that is, via student-blocked CV. This observation leads to a central question of this paper: Can we address this prediction fit limitation of the strong component model (AFM) without losing the explanatory power of the KC difficulty estimation?

3.3 Statistical Models

To fit the statistical models, we used a generalized linear mixed-effects model (lme4 package in R)[6] to specify both random and fixed effects parameters. All three models set student proficiency as a random effect and learning rate as a fixed effect, thus leaving the difficulty parameter as the discriminant for prediction.

3.3.1 Strong component Model(AFM)

Table 1: A comparison of three cross validation results (random, student-blocked & item-blocked) using root mean square error across three component models (strong, weak & hierarchical) for six datasets. The RMSEs in bold indicate the best predictive models.The hierarchical model is the best predictor for all six datasets for random and student-blocked CV and for 2 of 6 datasets for item-blocked CV. Small differences are seen in the remaining 4 datasets in the item-blocked CV between the strong and hierarchical models.

	Component Strong (AFM)			**Component Weak**			**Hierarchical Model**		
Data	Random	Student-blocked	Item-blocked	Random	Student-blocked	Item-blocked	Random	Student-blocked	Item-blocked
Geom	0.3972	0.4068	**0.3990**	0.3970	0.4055	0.4158	**0.3955**	**0.4051**	0.4009
Stats	0.3253	0.3287	**0.3357**	0.3169	0.3212	0.3493	**0.3153**	**0.3198**	0.3379
Math	0.4380	0.4917	**0.4405**	0.4159	0.4716	0.4645	**0.4157**	**0.4714**	0.4413
Eng1	0.4078	0.4409	0.4157	0.4027	0.4388	0.4160	**0.4026**	**0.4374**	**0.4150**
Eng2	0.3747	0.4017	0.3843	0.3609	0.3898	0.3890	**0.3604**	**0.3894**	**0.3819**
Eng3	0.3892	0.4179	**0.3939**	0.3841	0.4127	0.3988	**0.3836**	**0.4121**	0.3942

The strong component model, also known as the Additive Factors Model (AFM), is a logistic regression statistical model shown in R script in Equation 1. The response variable (correctness of student performance) is modeled as a function of the random effect for student proficiency combined with a fixed effect for knowledge component difficulty and a fixed effect for opportunity to practice each knowledge component. When KCs are modeled as fixed effects, the KC parameters estimates capture all the variance due to KC difficulty and there is no variance for items within the KC. The strong model uses parallel vectors of parameters with length equal to the number of KCs, thus explaining both difficulty and transfer using the same KCs.

$$correctness \sim (1|Student) + KC + KC : OppKC \tag{1}$$

3.3.2 Weak component Model(AFM')

The weak component model uses an item difficulty parameter to replace knowledge component difficulty found in the strong component model (see Equation 2). Unlike the strong model, the weak model provides a separate parameter for each item and, as such, does not provide a general explanation of difficulty, but merely a description of it. In this model, difficulty predictions (second term) are decoupled from transfer predictions (third term) as item is used for one and KC for the other.

$$correctness \sim (1|Student) + (1|Item) + KC : OppKC \tag{2}$$

3.3.3 Hierarchical Model(AFM'h)

The hierarchical model (AFM'h, Equation 3) models item difficulty through a hierarchical combination of KC-level and item-level estimates. Each item-level estimate is "pooled" within the KC it belongs and is thus constrained by the corresponding KC estimate. Item estimates are variations on the KC estimate and the model fit is penalized for item estimates away from zero (even as those estimates may improve correctness prediction). In machine learning terms, this constraint on item estimates is a kind of regularization. AFM'h provides an explanation of task difficulty in terms of knowledge components (as in AFM) but also provides an estimate of item difficulty (as in AFM').

$$correctness \sim (1|Student) + (1|Item/KC) + KC : OppKC \tag{3}$$

4. RESULTS

Figure 1a shows the root mean square error (RMSE) results from item-blocked CV across six datasets. The figure shows the weak model (see gray bars) is disadvantaged when generalizing across items as demonstrated by the height of the gray bars in comparison to the strong models (white bars) and hierarchical models (black bars). In all six datasets the weak models fared worse with item-blocked CV. Likewise, Figure 1b shows the RMSE for student-blocked CV where the component strong model (see white bars) fares poorly for all datasets in generalizing across students. These results confirm previous findings in which the strong model did better than the weak model in item-blocked CV (8 of 8 datasets) suggesting weak model has predictive disadvantages as well as explanatory ones[4]. Conversely, [4] found the weak model did better than the strong model in 7 of 8 datasets for student-blocked CV. Thus, there are predictive disadvantages of the strong model.

The important new result is that, while maintaining explanatory coherence and simplicity, the hierarchical model does not have either of these prediction disadvantages. See Table 1 for details. For both item-blocked and student-blocked CV, the prediction fits of the hierarchical model are never the worst. For student generalization (the student-blocked CV), 6 of 6 datasets have better prediction compared with the strong and the weak models and 2 of 6 for item generalization. They are essentially tied with the strong model in item generalization in the other cases. The differences slightly in favor of the strong model are in the Geom, Stats, Math, and Eng3 datasets. The pattern of results of random cross-validation, where data records are randomly assigned to CV folds irrespective of student or item tags, is highly similar to the pattern of results of student-blocked CV. Namely, the strong model is consistently the worst in prediction fit and the other two models are essentially tied.

5. DISCUSSION

This project advanced from the work done by [4] who provided clear evidence against the faculty theory, but identified an open issue about how best to implement the component theory. We hypothesized that combining the predictive power of a weak component model with the explanatory power of a strong component model would capture the best features of both models. Indeed from our results, the hierarchical models have as good or better prediction performance compared with the other two. The hierarchical model removes the prediction disadvantages of the strong model (for both student and item generalization) and both the explanatory disadvantage of the weak model and its prediction disadvantage on item generalization. Hence, we find that the hierarchical model gets the best of the both worlds: good prediction and good explanation.

One limitation of this study is that all the KC models used were single-KC models where each item is labeled by just one KC. Future work could attempt extend to models with multi-KC labeled items. Another future work possibility is to explore the use of the item random effect estimates of the hierarchical model as a better guide for KC model search than in recommended practice[9]. By comparing features of the hard items with those of the easy items, an analyst can hypothesize possible hidden skills and test whether associated KC relabeling produces better prediction fit. Current recommended practice relies on item means that may be biased estimates of item difficulty. These estimates are prone to inaccuracy particularly when there are a limited number of data points for

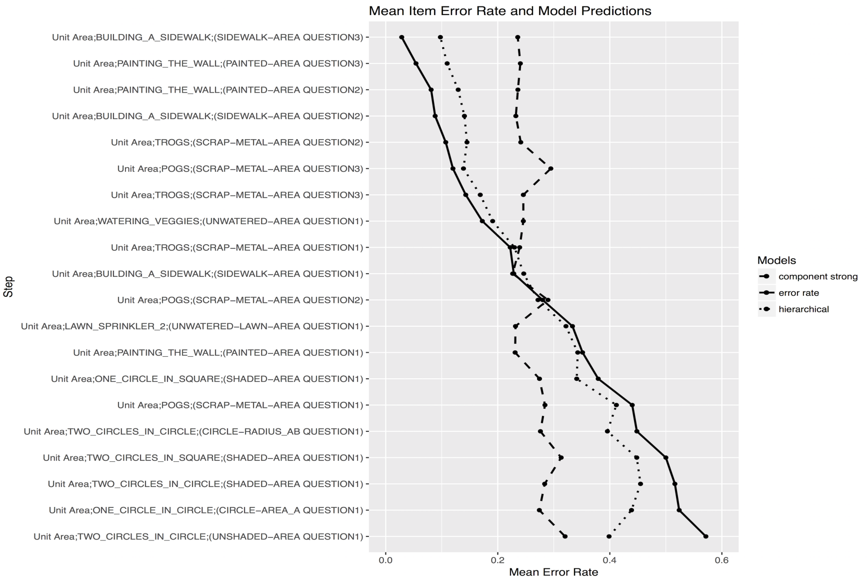


Figure 2: The hierarchical model (dotted line) better predicts item means (solid line) than the strong model (dashed line), but not perfectly so. Differences may indicate cases where the item mean is miss-estimating actual item difficulty perhaps because of limited data. An analyst trying to improve a KC model may be better guided by the hierarchical model predictions than by the simple means.

an item. This situation occurs frequently in early system design and testing, which is just the point in system development when KC model testing and improvement is most needed. Low item frequency also occurs in systems that automatically generate a wide variety of items (e.g., with random numbers in math). The hierarchical model provides a less biased estimate of item difficulty, which is more robust to small samples given that it is influenced by data on other items within the same KC. The KC estimate serves as a Bayesian prior for all items embedded within it.

Figure 2 shows a performance profiler displaying 20 steps/items labeled by a suspect KC in an early non-optimized KC model for the Geom dataset. The items are sorted by mean error rate as displayed in the solid line. In dashed line are the error predictions of the strong component model (AFM) which are particularly poor because an early and non-optimized KC model is being used. The hierarchical model predictions (in dotted) are closer to the mean error rate, but are importantly different. In particular, the last row in Figure 2 is the item with the highest mean error rate, but the hierarchical model suggests that it may not be so hard. Since analysts trying to improve a KC model are looking to identify possible knowledge demands that differentiate harder and easier items, it may be helpful for them to not be deceived by possible miss-estimates of difficulty resulting from using item means.

6. CONCLUSIONS

The search for highly predictive statistical models is a major focus of educational data mining and data mining more generally. This search is often pursued with less attention to the explanatory power of the models. Good explanatory models provide both scientific insight about the nature of learning and interpretable implications for improvement in educational interventions. This paper provides a model case, which we hope others will follow, of seeking a method that provides both predictive accuracy and explanatory power.

7. ACKNOWLEDGMENTS

This work was supported in part by a National Science Foundation grant (ACI-1443068). We thank Hui Cheng for help with programming and data analytics support.

8. REFERENCES

[1] S. M. Barnett and S. J. Ceci. When and where do we apply what we learn?: A taxonomy for far transfer. *Psychological bulletin*, 128(4):612, 2002.

[2] K. R. Koedinger, R. S. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. A data repository for the edm community: The pslc datashop. *Handbook of educational data mining*, 43, 2010.

[3] 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, 2012.

[4] K. R. Koedinger, M. V. Yudelson, and P. I. Pavlik. Testing theories of transfer using error rate learning curves. *Topics in cognitive science*, 8(3):589–609, 2016.

[5] R. Liu and K. R. Koedinger. Closing the loop: Automated data-driven cognitive model discoveries lead to improved instruction and learning gains. *Journal of Educational Data Mining*, 9(1):25–41, 2017.

[6] P. McCullagh and J. A. Nelder. *Generalized linear models*, volume 37. CRC press, 1989.

[7] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, pages 505–513, 2015.

[8] M. K. Singley and J. R. Anderson. *The transfer of cognitive skill*. Number 9. Harvard University Press, 1989.

[9] J. C. Stamper and K. R. Koedinger. Human-machine student model discovery and improvement using datashop. In *International Conference on Artificial Intelligence in Education*, pages 353–360. Springer, 2011.

[10] M. Wilson and P. De Boeck. Descriptive and explanatory item response models. In *Explanatory item response models*, pages 43–74. Springer, 2004.

Dynamic Knowledge Modeling with Heterogeneous Activities for Adaptive Textbooks*

Khushboo Thaker*, Yun Huang*, Peter Brusilovsky, Daqing He
School of Computing and Information
University of Pittsburgh
Pittsburgh, PA, USA
k.thaker,yuh43,peterb,dah44@pitt.edu

ABSTRACT

Adaptive textbooks use student interaction data to infer the current state of student knowledge and recommend most relevant learning materials. A challenge of student modeling for adaptive textbooks is that conventional student models are constructed based on performance data (quiz or problem-solving), however, students' interactions with online textbooks may produce a large volume of student reading data but a limited amount of performance data. In this work, we propose a dynamic student knowledge modeling framework for online adaptive textbooks, which utilizes student reading data combined with few available quiz activities to infer the students' current state of knowledge. The evaluation shows that proposed model learns more accurate students' knowledge state than Knowledge Tracing.

Keywords

student modeling, knowledge tracing, adaptive textbooks

1. INTRODUCTION

Adaptive online textbooks are one of the oldest technologies of personalized web-based learning [7, 10, 16]. A gradual shift to electronic books and textbooks over the last ten years makes this technology even more attractive than in its early days. The challenge for the modern research on adaptive textbooks is its integration with other online learning tools - problems, questions, animations, etc. In particular, student modeling (SM) approaches based on textbook readings behavior should be made compatible with more conventional SM based on student performance. This compatibility would support important "cross-content" recommendation where pages to read could be recommended through the analysis of problem-solving performance while interactive content (animations, problems, questions) could be recommended by considering the reading progress.

In performance-oriented intelligent tutoring systems (ITS), student knowledge state is measured on the level of individual domain skills or concepts, which are referred to as Knowledge Components (KCs). The main goal of KC-level knowledge modeling is to provide effective learning and reduce the total time of skill acquisition by offering adaptive feedback guiding the student to the most appropriate learning content. To support this personalization, the system keeps track of students' performance such as problem-solving and question-answering. These user interactions are later used by SM systems to distill student knowledge and predict student behavior.

Unfortunately, this well-explored approach could not be directly applied to adaptive textbooks. In most cases, textbook interaction logs provide only a small fraction of performance data (e.g., data on question answering and other activities related to course), which is not sufficient for timely and reliable SM. Naturally, these reading logs provide massive amount of data on student *reading*. However, the use of this data for SM is not straightforward because:

- The reading logs are noisy and not accurate. For example, a student can open a course content, start reading and then switch to some personal task.
- Individual differences (reading proficiency, motivation) could significantly affect student behavior.

In this paper, we present and evaluate a novel approach that combines student activities (reading data and performance data) to construct dynamic student knowledge model for adaptive textbooks. In the remainder of the paper, Section 2 discusses related work; Section 3 describes the proposed approach; Section 4 introduces the evaluation setup; Section 5 presents experimental results; and Section 6 summarizes conclusions and directions of future work.

2. RELATED WORK

2.1 Knowledge Tracing in ITS

Knowledge Tracing (KT) model was introduced in 1995 by Corbett and Anderson [3]. KT uses Hidden Markov Models (HMM) to represent student knowledge as binary latent variables. Each latent variable represents student knowledge of a particular KC, which could be either known or unknown. The observed variable is the performance of student at a given step, which is measured as a binary variable representing the correctness of a step or an answer (correct or not correct). KT directly represents KC-level knowledge estimation and allows dynamic knowledge update at each student learning opportunity. The conventional KT model

*both the authors contributed equally to the paper.(Does NOT produce the permission block, copyright information nor page numbering). For use with ACM_PROC_ARTICLE-SP.CLS. Supported by ACM.

has been extended further to learning individualized features [13] and providing instructional based intervention node [12]. In this work, we follow the KT modeling approach since we need knowledge estimates of different KCs to support several kinds of personalization.

2.2 Adaptive Online Textbooks

The research on adaptive textbooks has been motivated by the increasing popularity of World Wide Web (WWW) and the opportunity to use this platform for learning. The hypertext nature of early WWW made an online hypertext-based textbook a natural media for learning while the increased diversity of Web users stressed the need for adaptation. The first generation of adaptive textbooks [2, 4, 7, 10] focused on tracing student reading behavior to guide students to most relevant pages using adaptive navigation support [2, 4, 7, 16] or recommendation [10]. These types of personalization were based on a sophisticated knowledge modeling: each textbook page was associated with a set of concepts *presented* on the page as well as concepts *required* to understand the page [2, 4]. On the other hand, SM was relatively simple: these systems treated each visit to a page as a contribution to learning all presented concepts.

A significant trend of modern online textbooks is the increased inclusion of interactive content "beyond text". While the attempts to integrate online reading with problem solving have been made in the early days of online textbooks [16], it was a rare exception. Modern textbooks, however, routinely integrate a variety of "smart content" such as visualizations, problems, and videos. In this context, the ability to integrate data about student work with all these components and use it for a better-quality SM becomes a challenge for modern online textbooks.

3. KNOWLEDGE MODELING IN ADAPTIVE TEXTBOOKS

Our work attempts to combine the ideas of reading-based SM explored in the area of adaptive textbooks with the ideas of performance-based modeling explored by conventional ITS. The goal is to develop more reliable modeling for modern adaptive textbooks that could support several kinds of personalization such as guiding students to most appropriate sections or recommending relevant external content. This section introduces our earlier work on SM in textbooks and presents two novel models that combine reading-based KT [9] with performance-based KT [3] thus leveraging both reading and question-answering data.

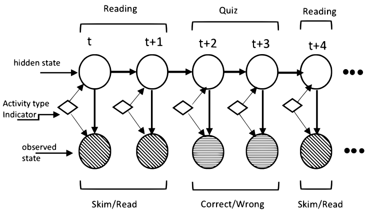
3.1 Behavior Model (BM) and Its Problems

As a baseline model in this work we use, Behavior Model (BM) suggested and explored earlier by Huang et al. [9]. The BM has a strict assumption that students reading speed is positively correlated with their knowledge state. However, other research indicated that this assumption might not always hold [1]. Indeed, in the dataset we considered for this study we observed a negative correlation between student reading behavior and quiz performance of -0.58, which indicates that data consists of mixture different types of students with noisy reading interactions. The primary goal of models presented in this paper was to improve BM. Our key ideas are (1) to handle mixture and noisy reading behavior among students by tuning it with other available activities performed by the student and (2) incorporate individual student differences to address better knowledge estimation for different types of students. In two following subsections, we present two models that advance the original BM in the proposed directions.

3.2 Behavior-Performance Model (BPM)

To achieve this we utilized Feature Aware Student Knowledge Tracing (FAST) framework [11], which replaces the conditional probability tables of the emission and transmission probabilities in BM framework with logistic regression (LR) distribution. HMM parameters are thus computed based on LR with features at each time step. This allows flexibility of incorporating a large number of features at each learning step. To enable FAST for different types of observation variables we introduce an activity type indicator variable which is set to 0 for *Read* and 1 for *Skim* (see Figure 1).

Figure 1: Behavior Performance Model (BPM)



3.3 Individualized Behavior-Performance Model (IBPM)

The *BPM* incorporates reading activities as binary variables with values *Skim* and *Read*. Since reading is a continuous variable, discretization of this manner causes a lot of information loss at student level. This information might be very helpful to characterize individualized student reading behavior and to obtain individualized parameters for different kinds of students. We propose *Individualized Behavior-Performance Model (IBPM)* that incorporates the individualized reading speed information as a feature in addition to activity type indicator features. This feature is based on accumulated median reading speed from first reading activity till $(t-1)$th reading activity of a student, where t is the current step of observation in an HMM of a KC. The feature is normalized to be in the range of 0 to 1 as there is a large variance in reading speed observation. Thus at each step along with different activity sequence observed, the model is also provided individual average reading speed observed so far. There are several benefits of our method:

- This method provides different sets of parameters (learn, guess, slip) for students with different reading speed.
- Compared with adding a parameter per-student for individualization, this feature provides more generalized modeling, because it learns the in-general association of the speed with HMM parameters for each KC.
- It is a flexible approach to integrate other behavior features as FAST has linear complexity in respect to the number of features [11].

4. EXPERIMENTS

4.1 System and Dataset

The dataset used for the experiment is collected from online reading platform Reading Circle [6] in spring 2016. This system was used for a graduate level course on Information Retrieval at the University of Pittsburgh. The system provides an active reading environment where students read the material of the assigned textbook to prepare for the next class. Each section of the assigned reading is followed by a quiz with several questions, which allow students to assess how well they learned the content. There is no restriction on the number of attempts to the questions. The final dataset contains 22,536 interactions from 22 students (see Table 1).

Table 1: Dataset Statistics

documents	394
questions	158
Average questions attempted	126
% of skimming Activities	33
% of reading Activities	67

4.2 Data-Preprocessing

Discretization of reading time is performed to label the observations to *Read* and *Skim*. For discretization we followed the same technique as performed by Huang et al. [9] The key to well-trained KT model is to have correct representative KCs. The conventional way of defining KCs is manual knowledge modeling by subject experts. Recently, Huang et al. [9], tried different KC extraction methods and found automatic word-based method to be reliable. However, word-based method gives a large set of KCs and it is very noisy. To improve automatic KC extraction based on words' importance in a reading unit, we applied the TF*IDF (Term Frequency - Inverse Document Frequency) approach. For each document, top 5 TF*IDF-weighed words were extracted and considered as KCs for that reading.

4.3 Tools and Parameters

For building both BPM and IBPM models, we used open source FAST toolkit [5]. HMM models are prone to get trained for local optimum values, due to which proper initialization of HMM parameters is very important. In all the models the HMM modes were initialized with (0.1,0.1,0.8,0.8) parameter values for ($P(L_0)$,$P(T)$,$P(G)$,$P(S)$). This choice of initialization is based on observing the negative correlation between reading and performance and preliminary experiments under another initial parameter set (0.1,0.1,0.2,0.2) where the predictive performance of all models was worse [9].

4.4 Baseline Methods

In order to show the performance gain of proposed approach, we used two variations of KT as baselines. The first model is the *Behavior Model* (**BM**) reviewed in section 3.1, and the second is *Performance Model* (**PM**) trained on quiz activities by the student. In addition we use a majority class baseline (**MC**). As the proposed model is able to perform both reading time and quiz performance predictions, **BM** and **PM** separately act as a baseline for proposed models' reading time prediction and quiz performance prediction task.

4.5 Cross Validated Prediction Evaluation

FAST trains individual HMM for each KC using training data and performs prediction on test data. Firstly, we randomly selected 50% of students and put all their reading and quiz activity data into training set. Then for the remaining 50% of students, we put the first half of their activity sequence into training set. The second half of their activity sequences are withheld for test set. This process is repeated 10 times. The prediction is reported on reading speed, first attempt quiz performance, and all-attempts quiz performance. 10 split cross-validation is performed from the generated folds. Both Area Under the Receiver Operating Characteristic curve (AUC) and Root Mean Squared Error (RMSE) are reported based on a recent paper, that raised a concern about using only AUC for evaluation of SM [14].

5. RESULTS AND DISCUSSION

5.1 Predictive Performance of BPM

Table 2 summarizes the predictive performance computed by averaging across 10 splits and Table 3 reports significance. Comparing with *MC*, *BPM* has significantly better RMSE and AUC across all prediction tasks. The relatively lower AUC value of *BPM* in reading prediction task indicates high noise in reading interactions. Since quiz performance usually correlates better with knowledge than reading behavior, the prediction on quiz is of more importance than that on reading, thus the result indicates a clear advantage of *BPM* over *MC*. Comparing with *BM* and *PM* which are trained on a single type of interactions, *BPM* also beats them significantly in corresponding prediction tasks in both RMSE and AUC metrics. We clearly see the advantage of integrating behavior and performance data in *BPM* over *PM* and *BM*. Better performance of *BPM* over *BM* indicates that even a small amount of quiz performance data could significantly improve knowledge inference and performance prediction. Better performance of *BPM* over *PM* indicates that reading data albeit being noisy still carries valuable information that could help infer knowledge and conduct prediction.

5.2 Predictive Performance of IBPM

The intuition behind *IBPM* is that it provides additional student reading behavior features (in addition to activity type indicator) for capturing individual differences. As can be seen in Table 2, *IBPM* incorporating individualized speed feature shows improvement by both RMSE and AUC metrics compared with *BPM*. The improvement is significant for reading speed prediction task and quiz all-attempts performance prediction. However, its improvement over *BPM* on predicting first attempt performance in terms of RMSE is not significant. A probable reason is that our dataset exhibits a mixture of students in terms of reading behavior and performance (indicated by negative correlation value).

Table 2: Prediction performance for reading speed, 1st attempt quiz prediction, and all attempts. Two best results are shown in bold.

Model	RMSE	AUC	RMSE	AUC	RMSE	AUC
	reading		1st att.		all att.	
IBPM	**.483**±.008	**.512**±.014	**.472**±.004	**.635**±.018	**.391**±.007	**.867**±.010
BPM	**.487**±.008	.458±.012	**.473**±.004	**.633**±.018	**.391**±.007	**.867**±.010
BM	.508±.011	.442±.019	-	-	-	-
PM	-	-	.504±.002	.602±.014	.427±.005	.803±.009
MC	.593±.019	**.500**±.000	.550±.013	.500±.000	.693±.003	.500±.000

Table 3: Paired t-test p value for reading and quiz prediction performance with Bonferroni correction

	read		1st att.		all att.	
Compared Models	RMSE	AUC	RMSE	AUC	RMSE	AUC
IBPM vs *BPM*	***	***	0.18	*	*	*
IBPM vs *BM/PM*	***	***	***	***	***	***
IBPM vs *MC*	***	**	***	***	***	***
BPM vs *BM/PM*	***	***	*	***	***	*
BPM vs *MC*	***	***	***	***	***	***

10CV paired t-test, p-values
*0.05/5 = 0.01, **0.01/5 = 0.002, ***0.001/5 = 0.0002

5.3 Parameter Analysis of BPM

To validate our hypothesis that quiz activities contain less noise than reading activities for inferring knowledge, we conduct a drill-down analysis of parameters of *BPM* and baseline models. We compute the parameters for each KC in

BPM by setting the value of activity type indicator to 0 for the reading part and 1 for quiz part in the logistic regression of each parameter, and then average the parameters across all KCs. According to Table 4, BPM has fitted lower *guess* and *slip* parameters in quiz activity part than reading activity part, which indicates that quiz activities have higher positive correlation with knowledge state than reading activities i.e., quiz activities indeed have much less noise for inferring knowledge. In addition, Table 4 shows that the parameters learned for guess and slip for BPM are smaller than those for BM and PM, which indicates that BPM has higher plausibility enabling more accurate knowledge inference than these baseline models [8]. The high values of *guess* and *slip* parameters for BM and PM model indicates that single activity is not able to learn accurate student behavior.

Table 4: Parameters learned by different models for learn, guess and slip probabilities

Model	Activity Type	learn	guess	slip
BM	Reading	0.384	0.505	0.776
PM	Quiz	0.091	0.705	0.589
BPM	Reading	0.404	0.363	0.420
BPM	Quiz	0.354	0.288	0.313

6. CONCLUSION AND FUTURE WORK

This paper investigated the significance of integrating heterogeneous student activities in a KT framework for adaptive textbooks. The integrated model BPM was trained with large volume of noisy reading data and small amount of quiz performance data. BPM significantly outperforms the basic model BM, which is based on only reading behavior logs, and PM which is based on only quiz behavior logs. The results indicate that combining quiz and reading interactions help in inferring student knowledge state. To address student differences, $IBPM$ integrated continuous observation in BPM. The performance of $IBPM$ was similar to BPM with a considerable improvement on reading speed prediction and small improvement on quiz performance prediction. In the future, we would like to further investigate $IBPM$ by utilizing other individualization features.

Although overall performance is not as high as in ITS focused on mastery learning, our past experience with topic-based SM [15] hints that current level of prediction performance could be sufficient to deliver successful personalization based on adaptive navigation support where the student can choose from several recommended options. We plan to assess the value of our SM approach as a basis for personalized guidance in the future studies.

Our work could be considered as the first attempt to model dynamic student knowledge in adaptive textbooks with heterogeneous interactions. We believe that the possibility of integrating individual differences to the proposed model makes it especially promising for real-time learning systems. Moreover, our approach makes it possible to integrate more types of student activities like search, video, listening and discussion to further increase the quality of modeling and to provide holistic SM. We plan to explore these opportunities in the future work.

7. REFERENCES

[1] R. S. Baker, A. T. Corbett, K. R. Koedinger, and A. Z. Wagner. Off-task behavior in the cognitive tutor classroom: When students ”game the system”. In *Proc. ACM Conf. on Human Factors in Computing Systems*, CHI ’04, pages 383–390, 2004.

[2] P. Brusilovsky and J. Eklund. A study of user-model based link annotation in educational hypermedia. *J. of Universal Computer Science*, 4(4):429–448, 1998.

[3] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modelling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1995.

[4] P. De Bra. Teaching through adaptive hypertext on the www. *Int. Journal of Educational Telecommunications*, 3(2/3):163–180, 1997.

[5] J. González, Y. Huang, and P. Brusilovsky. General features in knowledge tracing: Applications to multiple subskills, temporal item response theory, and expert knowledge. In *Proc. the 7th Int. Conf. on Educational Data Mining*, pages 84–91, 2014.

[6] J. Guerra, D. Parra, and P. Brusilovsky. Encouraging online student reading with social visualization. In *The 2nd Workshop on Intelligent Support for Learning in Groups at the 16th Conf. on Artificial Intelligence in Education*, pages 47–50, 2013.

[7] N. Henze, K. Naceur, W. Nejdl, and M. Wolpers. Adaptive hyperbooks for constructivist teaching. *Künstliche Intelligenz*, 13(4):26–31, 1999.

[8] Y. Huang, J. González, R. Kumar, and P. Brusilovsky. A framework for multifaceted evaluation of student models. In *Proc. the 8th Int. Conf. on Educational Data Mining*, pages 203–210, 2015.

[9] Y. Huang, M. Yudelson, S. Han, D. He, and P. Brusilovsky. A framework for dynamic knowledge modeling in textbook-based learning. In *Proc. 24th Conf. on User Modeling, Adaptation and Personalization*, pages 141–150, 2016.

[10] A. Kavcic. Fuzzy User Modeling for Adaptation in Educational Hypermedia. *IEEE Transactions on Systems, Man and Cybernetics*, 34(4):439–449, 2004.

[11] M. Khajah, Y. Huang, J. González, M. C. Mozer, and P. Brusilovsky. Integrating knowledge tracing and item response theory: A tale of two frameworks. In *Workshop on Personalization Approaches in Learning Environments at Int. Conf. on User Modeling, Adaptation, and Personalization*, pages 7–12, 2014.

[12] C. Lin and M. Chi. Intervention-BKT: Incorporating instructional interventions into bayesian knowledge tracing. In *Intelligent Tutoring Systems*, pages 208–218, Cham, 2016. Springer.

[13] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *Proc. the 18th Int. Conf. on User Modeling, Adaptation, and Personalization*, pages 255–266. Springer-Verlag, 2010.

[14] R. Pelánek. Metrics for evaluation of student models. *Journal of Educational Data Mining*, 7(2):1–19, 2015.

[15] S. Sosnovsky and P. Brusilovsky. Evaluation of topic-based adaptation and student modeling in quizguide. *User Modeling and User-Adapted Interaction*, 25(4):371–424, 2015.

[16] G. Weber and P. Brusilovsky. ELM-ART: An adaptive versatile system for web-based instruction. *Int. Journal of Artificial Intelligence in Education*, 12(4):351–384, 2001.

Contextual Derivation of Stable BKT parameters for Analyzing Content Efficacy

Deepak Agarwal
Educational Initiatives
Ahmedabad, IN
+91 9731406003
deepak.agarwal@ei-india.com

Nishant Babel
Educational Initiatives
Ahmedabad, IN
+91 7022281740
nishant.babel@ei-india.com

Ryan S. Baker
3700 Walnut St.
Philadelphia, PA
+1 (412) 983-3619
ryanshaunbaker@gmail.com

ABSTRACT

One of the key benefits that Bayesian Knowledge Tracing (BKT) offers compared to many competing student modelling paradigms is that its parameters are meaningful and interpretable. These parameters have been used to answer basic research questions and identify content in need of iterative improvement (due to, for instance, low learning or high slip rates). However, a core challenge to the interpretation of BKT parameters is that several combinations of BKT parameters can often fit the same data comparably well. Even if, as some have argued, BKT is not truly non-identifiable, in practice highly different parameters with comparable goodness are often found using modern BKT fitting packages. These parameter sets can have highly divergent values for guess and slip. Several approaches have been proposed but none of those have yet led to fully stable and trustworthy parameter estimates. In this work, we propose a new iterative method based on contextual guess and slip estimation that converges to stable estimates for skill-level guess and slip parameters. This method alternates between calculating contextual estimates of guess and slip and estimating skill-level parameters, iterating until convergence. Thus, it produces a more stable set of parameters that can be more confidently used in analyzing content efficacy.

Keywords

Bayesian Knowledge Tracing, Contextual Guess Slip, Content Efficacy, Brute Force BKT Model, EDM

1. INTRODUCTION

The process of developing an intelligent tutoring system (ITS) is an iterative one, and content frequently needs revision to reach its desired effectiveness for students [1]. In addition, even as intelligent tutors have become more widespread, the quality of the content present in them has often become more varied, with the advent of approaches such as crowd-sourcing for generating large amounts of content quickly [2]. As such, improving the quality of content that the students are exposed to is one of the significant aspects of the development of ITS. One approach to achieving this is to put in place a framework for automatically reviewing content, identifying/flagging content that does not meet the desired objectives. In this paper, we discuss our efforts to create such a system within the context of Mindspark*, an ITS software being used by over 80,000 students in India.

The cornerstone of our efforts is discovering skills which have unexpected negative properties, specifically very low learning rates, or very high rates of guess and slip within the Bayesian Knowledge Tracing paradigm (Corbett & Anderson, 1995). Bayesian Knowledge Tracing is a highly-cited paradigm for student modeling and is used in a wide range of real-world adaptive learning systems. Although recent evidence suggests that extensions to BKT and competing paradigms may in some cases achieve better prediction of immediate correctness [3], BKT remains a high-quality, highly interpretable paradigm for modeling student latent knowledge as well as meaningful attributes of individual skills. However, one challenge to interpreting BKT parameters is that different sets of parameters can fit the data comparably well [4]. Although recent articles have argued that BKT is not truly non-identifiable [5], nonetheless contemporary packages for choosing BKT parameters regularly produce very different parameter values with comparable fits. Other researchers have noted the problem of unstable parameters; however, these approaches have tended to assume skills have similar parameters to each other [6], [7]. These assumptions may lead to more plausible parameters in general but may be unhelpful for identifying skills whose guess, slip, or learning rates are genuinely problematic.

Hence, in this paper we propose a new iterative approach for stabilizing the parameter values of BKT that leverages additional information about student performance. Previous work proposed contextually estimating guess and slip in all cases with situational information [8]; this approach produced unstable improvements in model goodness, however, with positive impacts in some data sets and negative impacts in other data sets. This paper instead uses iterative contextual estimation of guess and slip to help select guess and slip parameters for traditional Bayesian Knowledge Tracing – i.e. the final model is non-contextual. By using additional information to derive better estimates of guess and slip, we can be more confident about our model parameters, and more confident about our ability to use these parameters in driving quality improvement.

The subsequent sections explain the dataset used, the conceptualization of the iterative parameter estimation approach, the results obtained and how we validated the approach. We conclude with a discussion of how this approach will be leveraged, going forward, to analyze content efficacy in Mindspark.

2. DATA DESCRIPTION AND APPROACH

2.1 Data sets

In order to evaluate our approach to estimate BKT parameters, we considered a simulated data set and a genuine data set from the Mindspark platform.

*https://www.mindspark.in/

Simulated data:

Student responses were simulated for four different skills by assuming four different set of BKT model parameters values – L_0, G, S, T – and then estimating the probabilities of student responses using the BKT paradigm. Each skill had 2,000 users, with four attempts for every user. Each of the four skills came from a different [L_0, G, S, T] combination: [0.6, 0.3, 0.05, 0.25], [0.6, 0.3, 0.05, 0.02], [0.6, 0.05, 0.25, 0.25] and [0.6, 0.05, 0.25, 0.02] respectively. Data consisted of 0 and 1 for incorrect and correct response by users on each attempt. The data was simulated by calculating the likelihood of knowledge and a correct response based on the BKT model and each simulated user's response history.

Real student log data from Mindspark Math:

Actual student response data was also extracted from the Mindspark log data. Mindspark is an adaptive-learning program for Math and English, developed by Educational Initiatives (EI). Mindspark Math currently has 80,000 users, primarily from private schools, in grades 1 to 9, across India. For our purpose, data from the 'Revision Module' in Mindspark was taken. The Revision Module is a 30-minute session that gives students questions from topics selected by the teacher. It is intended to help students learn concepts for which that student had relatively low performance in regular modules. The reason for selecting the Revision Module for this exercise was that this module usually has multiple attempts per skill for each student. A ′attempt′ here means an opportunity provided to a student to apply the skill in order to solve a question. Hence when a student has multiple attempts on a particular skill, the student is presented a set of questions testing the same skill and each response is scored as correct/incorrect. Data of 1,032 users across a total of 5,200 attempts was extracted for six different skills. We limited the data set to students who had at least three attempts and capped the number of attempts at five per user in order to avoid focusing the data set on students who struggled to reach mastery.

Skill	Grade	Learning Objective
NTH001_8	5	Determining least multiple for a number out of a given set of numbers
WNC034_7	5	Estimating a number to the nearest hundred
NTH021_8	5	Writing factorizations of a number using the factor tree of the number
WNC059_12	5	Estimating a number to the nearest thousand
WNO033_10	5	Adding a 3-digit number to another 3-digit number vertically
WNO049_10	5	Writing quotient and remainder given dividend and divisor

Table 1. Skills used from the Revision Module of Mindspark

2.2 Approach

We derived contextually-inspired parameters for Bayesian Knowledge Tracing as follows: We start by obtaining initial parameter values for each skill using the common Brute Force grid search method [9] and classical BKT paradigm. This set of parameter values are used as input to the Contextual Guess Slip model [8] to estimate the contextual probability of guess and slip for each student attempt. The contextual probability of guess and slip is derived from the likelihood of a student knowing the skill at a specific attempt, which in turn is estimated based on the student's performance on the next two attempts on that skill. The formulas from the original contextual guess/slip model [8] were used to calculate $P(L_{n-1})$ which represents each student's knowledge state after (n-1) th attempt. The formulas take into account a student's subsequent two attempts (n and n+1 response data) to calculate $P(L_{n-1})$.

$$P(L_{n-1} \mid A_{n,n+1}) = P(A_{n,n+1} \mid L_{n-1}) * P(L_{n-1}) / P(A_{n,n+1}) \quad (1)$$

$$P(A_{n,n+1}) = P(L_{n-1}) * P(A_{n,n+1} \mid L_{n-1})$$

$$+ (1 - P(L_{n-1})) * P(A_{n,n+1} \mid \sim L_{n-1}) \quad (2)$$

The probability of the actions at time n and n+1, in the case that the student knew the skill at time n (L_{n-1}), is a function of the probability that the student guessed or slipped at each opportunity to practice the skill. C denotes a correct action; ~C denotes an incorrect action.

$$P(A_{n,n+1} = C,C \mid L_{n-1}) = P(\sim S)^2 \quad (3)$$

$$P(A_{n,n+1} = C,\sim C \mid L_{n-1}) = P(\sim S)*P(S) \quad (4)$$

$$P(A_{n,n+1} = \sim C,C \mid L_{n-1}) = P(S)*P(\sim S) \quad (5)$$

$$P(A_{n,n+1} = \sim C,\sim C \mid L_{n-1}) = P(S)^2 \quad (6)$$

The probability of the actions at time n and n+1, in the case that the student did not know the skill at time n ($\sim L_{n-1}$), is as below:

$$P(A_{n,n+1} = C,C \mid \sim L_{n-1}) = P(G)*P(\sim T)*P(G)$$

$$+ P(G)*P(T)*P(\sim S) \quad (7)$$

$$P(A_{n,n+1} = C,\sim C \mid \sim L_{n-1}) = P(G)*P(\sim T)*P(\sim G)$$

$$+ P(G)*P(T)*P(S) \quad (8)$$

$$P(A_{n,n+1} = \sim C,C \mid \sim L_{n-1}) = P(\sim G)*P(T)*P(\sim S)$$

$$+ P(\sim G)*P(\sim T)*P(G) \quad (9)$$

$$P(A_{n,n+1} = \sim C,\sim C \mid \sim L_{n-1}) = P(\sim G)*P(T)*P(S)$$

$$+ P(\sim G)*P(\sim T)*P(\sim G) \quad (10)$$

After calculating $P(L_{n-1})$, the contextual probabilities of guess and slip at nth attempt was assigned as:

$$P(G'_n) = 1 - P(L_{n-1}) \quad (11)$$

$$P(S'_n) = P(L_{n-1}) \quad (12)$$

The probabilities obtained are at a student attempt level. To obtain skill level parameters, we aggregate these values across all the attempts for a given skill as below:

$$G = \Sigma\, P(G'_n \mid C) / \Sigma\, P(G'_n) \quad (13)$$

$$S = \Sigma\, P(S'_n \mid \sim C) / \Sigma\, P(S'_n) \quad (14)$$

where $P(G'_n)$ and $P(S'_n)$ are taken from equations 11 and 12 respectively.

In other words, to obtain the guess parameter, we take the ratio of the sum of the $P(G'_n)$ values for the attempts where the response by the student was correct and the sum of the $P(G'_n)$ values across all the attempts for the skill. Similarly, to obtain the slip parameter, we take the ratio of sum of the $P(S'_n)$ values for attempts where the response was incorrect and the sum of $P(S'_n)$ values across all the attempts for the skill.

Subsequently, if the skill level guess and slip estimates from the contextual model do not agree with the guess and slip estimates obtained from the Brute Force grid search method originally, then it means that the BKT parameters are not stable. These parameters can be refined by using contextual estimates of guess and slip as input to Brute Force grid search algorithm and iterating this process until the skill level G and S values from the two approaches match with each other. Here note that each iteration is performed on the entire dataset (all the attempts of the students) which means the dataset does not change from one iteration to another. The flowchart below summarizes the whole process:

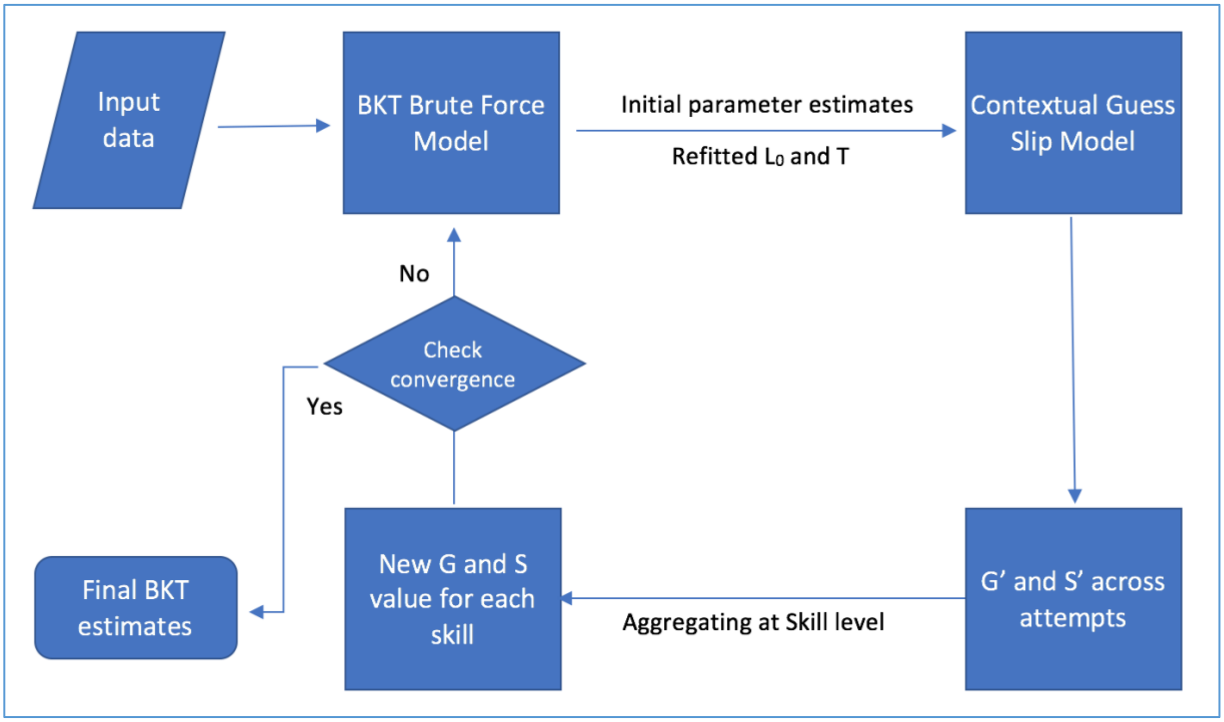

Input data
BKT Brute Force Model
Initial parameter estimates
Refitted L₀ and T
Contextual Guess Slip Model
No
Check convergence
Yes
Final BKT estimates
New G and S value for each skill
Aggregating at Skill level
G' and S' across attempts

Fig. 1. BKT with Contextual Guess Slip Flow

3. VALIDATION AND RESULTS

3.1 Validations carried out on simulated data

The purpose of using simulated data was to confirm that the skill level G and S values calculated through the contextual model match the original G and S values used to generate the simulated data. The calculated values from the proposed approach achieved a match within 1% error margin to the parameter values used for simulating the data for all four cases.

Table 2. Original vs Calculated G and S values for each skill

Data set	Original G	Original S	Calculated G	Calculated S
Skill 1	0.3000	0.0500	0.2998	0.0499
Skill 2	0.3000	0.0500	0.3001	0.0500
Skill 3	0.0500	0.2500	0.0500	0.2501
Skill 4	0..0500	0.2500	0.0500	0.2501

We also used the simulated data to check if the iterative model achieves convergence over time and results in a stable set of parameter values. For this purpose, we used arbitrary parameter values [L_0=50%, G=15%, S=15%, T=10%] for first iteration for all four simulated skills instead of estimating the parameters from the Brute Force BKT model. We observed that L_0, G, S, T values started converging after a reasonable number of iterations for all four cases and the output matched the original parameter set used to simulate the data. Fig. 2 shows the convergence for all four simulated skills. We have also shown the trend in RMSE values over the iterations for all four skills in Fig 4. Since we had started with arbitrary parameter values, RMSE is quite high in the first iteration but decreases continuously to achieve the minimum value over multiple iterations.

3.2 Validation carried out on Mindspark data

It is not possible to determine whether the proposed approach reaches "true" parameter values for real-world data, because it is unknown what those true parameter values would be (and, indeed, we know that BKT is an imperfect model of the real world). However, we are still able to validate how well the proposed approach converges when applied to real data, where the noise may be different in kind than the noise generated by BKT for the simulated data. As Fig. 3 shows below, the curve for all four skills starts flattening out after a tractable number of iterations, exhibiting convergence in the data.

We also show here that the model achieves convergence without significantly increasing the original RMSE achieved through the Brute Force BKT model which indicates that the model fit does not worsen over the iterations. The change in the RMSE values was observed to be under 0.002 across all six skills and the trend has been shown in Fig 4.

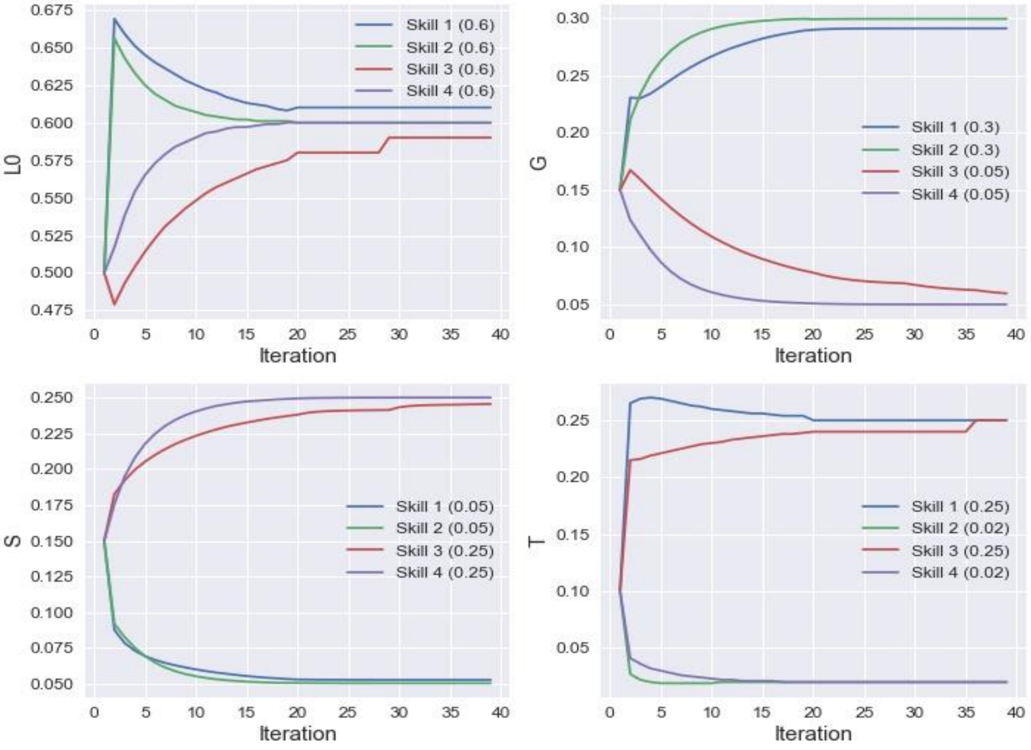


Fig. 2. Contextual BKT approach on the simulated data with four skills across 40 iterations. The values in the parentheses represent the actual values used for simulation

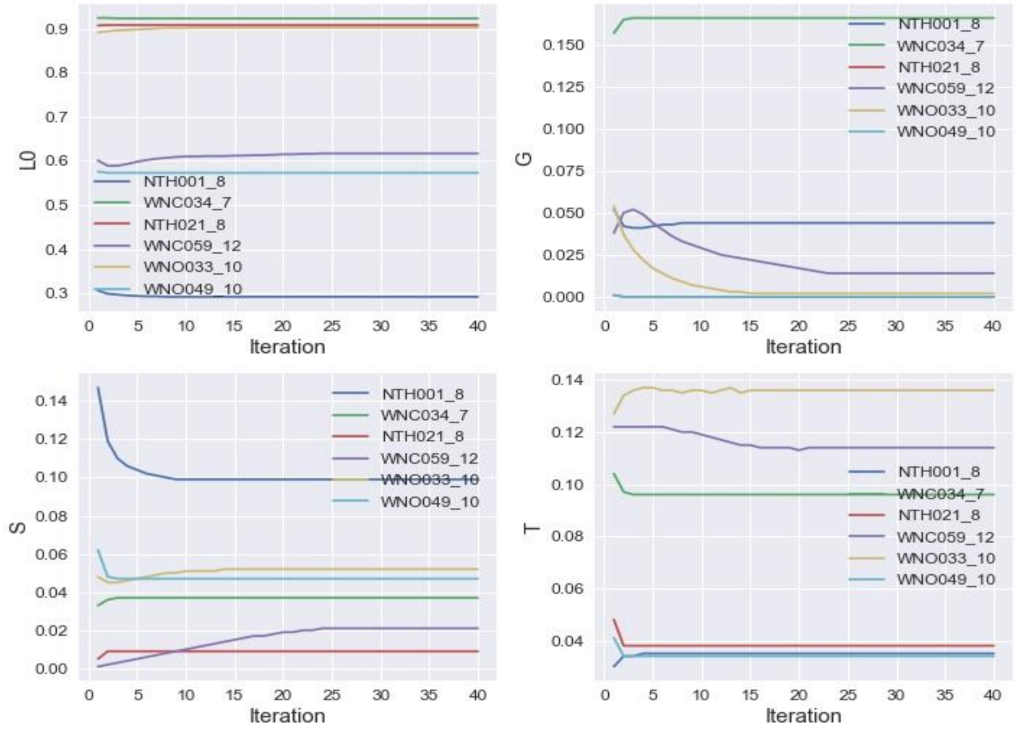


Fig. 3. Contextual BKT approach on Mindspark data with six skills across 40 iterations

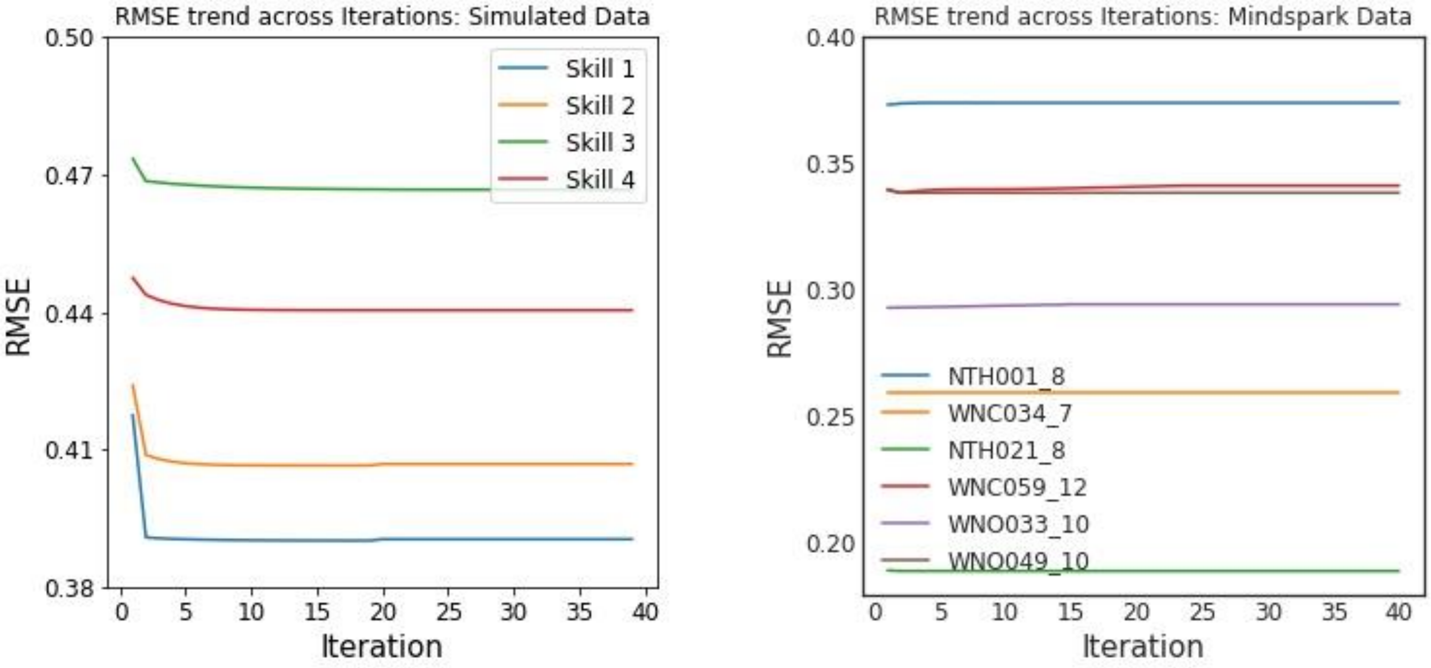


Fig. 4. RMSE values of the Contextual BKT model for simulated and real data across 40 iterations. As can be observed, the error metric changes very minimally across iterations.

4. CONCLUSION AND FUTURE WORK

In this paper, we discuss a new, iterative approach to fitting BKT parameters, involving iteration between fitting skill-level parameters for guess and slip, and contextually estimating guess and slip within each problem attempt. This approach converges across iterations to a stable and single set of parameters which have a principled justification for their selection. As such, we can then have higher confidence about interpreting and using these parameters for commenting on content efficacy. The skill level BKT parameters enable us to evaluate content on multiple dimensions: (a) grade appropriateness, (b) learning rate, and (c) quality of the content as indicated by low or high guess and slip values. For content to be effective in a given context, it should have BKT parameters within a desired range. In future work, the desired range of values will be determined through multiple approaches including analyzing parameter distributions to set up heuristic rules, anomaly detection, and discussions with our pedagogical experts.

If the parameter values for a skill is outside of those permissible ranges, it would indicate that the content does not meet the quality/effectiveness standard. For example, for a piece of content to be grade appropriate, L_0 should likely be between 25% to 85%. Any content which has L_0 above 85% may not lead to substantial improvement in student learning, as most of the students already know it. By contrast, any content which has L_0 below 25% is also not appropriate as students may not know the pre-requisite skills to learn the content.

Our next step is therefore to establish thresholds (the desired range of values) for each parameter to develop filters which will automatically identify ineffective content and bring it to the attention of the content developers.

An added advantage of screening content using BKT parameters is that we can also provide auto-generated guidance on what the issue might be with the content rather than just highlighting that the content needs improvement. Apart from being a tool for ITS / content developers in identifying lower quality content for revision, this approach has wider application in the domain of Bayesian Knowledge Tracing as it provides a means to capture a single set of skill parameters and have a justification for preferring this set of values to others with comparable fit.

5. REFERENCES

[1] Koedinger, K. R., Stamper, J. C., McLaughlin, E. A., & Nixon, T. (2013). Using data-driven discovery of better student models to improve student learning. *Proc. International Conference on Artificial Intelligence in Education*, 421-430.

[2] Heffernan, N. T., Ostrow, K. S., Kelly, K., Selent, D., Van Inwegen, E. G., Xiong, X., & Williams, J. J. (2016). The Future of Adaptive Learning: Does the Crowd Hold the Key? *International Journal of Artificial Intelligence in Education*, *26*(2), 615-644

[3] Khajah M., Lindsey, R.V., Mozer, M. (2016) How deep is knowledge tracing? *Proc. Int'l. Conf. on Educational Data Mining*, 94-101

[4] Beck, J. E., & Chang, K. M. (2007, July). Identifiability: A fundamental problem of student modeling. In *International Conference on User Modeling* (pp. 137-146). Springer, Berlin, Heidelberg.

[5] Doroudi S., Brunskill E. (2017). The Misidentified Identifiability Problem of Bayesian Knowledge Tracing. *Proc. Int'l. Conf. on Educational Data Mining.*

[6] Rai, D., Gong, Y., & Beck, J. E. (2009) Using Dirichlet priors to improve model parameter plausibility. *Proc. Int'l. Conf. on Educational Data Mining*, 141-150.

[7] Ritter, S., Harris, T. K., Nixon, T., Dickison, D., Murray, R. C., & Towle, B. (2009). Reducing the Knowledge Tracing Space. *Proc. Int'l. Conf. on Educational Data Mining.*

[8] 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. *Proc. Int'l. Conf. on Intelligent Tutoring Systems.*

[9] Baker, R.S.J.d., Corbett, A.T., Gowda, S.M., Wagner, A.Z., et al. (2010) Contextual Slip and Prediction of Student Performance After Use of an Intelligent Tutor. *Proceedings of the 18th Annual Conference on User Modeling, Adaptation, and Personalization*, 52-63.

Constructing Cognitive Profiles for Simulation-Based Hiring Assessments

Rebecca Kantar

Imbellus
1085 Gayley Ave.
Westwood, CA 90024
rebecca@imbellus.com

Keith McNulty

McKinsey & Company
1 Jermyn St, S. James
London, UK
Keith_mcnulty@mckinsey.com

Erica L. Snow

Imbellus
1085 Gayley Ave.
Westwood, CA 90024
esnow@imbellus.com

Matthew A. Emery

Imbellus
1085 Gayley Ave.
Westwood, CA 90024
memery@imbellus.com

Richard Wainess

Imbellus
1085 Gayley Ave.
Westwood, CA 90024
rwainess@imbellus.com

Sonia D. Doshi

Imbellus
1085 Gayley Ave.
Westwood, CA 90024
sdoshi@imbellus.com

ABSTRACT

Imbellus is an assessment company that builds immersive simulation-based assessments designed to evaluate cognitive processes. The work described here explores our partnership with McKinsey & Company, a best-in-class management-consulting firm, to build a simulation-based assessment that evaluates incoming applicants' cognitive skills and abilities. Our simulation-based assessments are designed to produce a substantial amount of information about the incoming applicants, including metacognitive skills, decision-making processes, and situational awareness (to name a few of the constructs we measure). This paper will explore the rich telemetry data we collect and quantify, as well as the novel scoring and exploratory techniques we are conducting to gain insight into applicants' cognitive profiles. We will present our initial findings and describe implications of our current work for the fields of artificial intelligence, educational data mining, and assessment.

Keywords

Cognitive Assessment, Learning Science, Machine Learning

1. INTRODUCTION

Imbellus assessments are designed to provide a wealth of information concerning applicants' cognitive skills and profiles. In contrast, traditional standardized cognitive assessments primarily evaluate content mastery, processing speed, and memory. The rise of automation makes insights around domain knowledge, processing speed, and memory less relevant features of human cognition, while higher level, complex cognitive abilities become features that make all the difference in individuals' preparedness for modern work and life. Imbellus assessments evaluate what have historically been hard-to-measure skills like problem-solving, creativity, systems thinking, and critical thinking. To take a practical approach to designing good assessments, Imbellus partners with industry leaders whose employees leverage key 21st Century skills at an elite level. Our early work with McKinsey & Company, a best-in-class management-consulting firm, has involved building an assessment to gauge incoming applicants' cognitive skills and abilities, which will be used to construct profiles of each applicant.

Standardized cognitive assessments were developed in the late 1800s to "stratify students of different abilities into different curricular paths" [9]. The release of Goddard's IQ formula and the Stanford-Binet cognitive assessment in the early 1900s launched a movement of mass testing in the United States. The College Entrance Examination Board, now the College Board, was established in 1923 to define a set of college admission standards through the dissemination of the Scholastic Aptitude Test (SAT) [3]. In 1959, the American College Test (ACT) was released as an alternative to the SAT [3]. The ACT's stated goal is to "measure information taught in high school," instead of evaluating cognitive reasoning skills. [8]. The ACT and SAT set college admissions standards, which became significant shaping forces. Today over 39 Advanced Placement tests and 20 SAT Subject tests dictate the curriculum in our K-12 education system and influence infrastructure and resource allocation. The ACT and the SAT focus on standardized content in mathematics, writing, science, and other subject-specific areas to create objective metrics. [6]. While widely adopted across the nation, these assessments have "revealed little about specific cognitive abilities or predicted performance" [3].

In response to the shortcomings in both the methodology of and substance of traditional standardized college admissions tests, employers have adopted other traditional cognitive ability or intelligence tests in an effort to glean more predictive insights on applicants' cognitive profiles. Most cognitive ability tests measure "reasoning, perception, memory, verbal and mathematical ability" [1]. These assessments, like standardized admissions tests, focus on content mastery, processing speed, and memory. These factors ignore the increasing need to develop and measure capabilities required by the 21st-century workforce. These tests ignore the cognitive process that users engage in during that task.

Past their shortcomings in predictive validity, most cognitive assessments are paper-and-pencil multiple-choice tests, a medium for evaluating cognitive skills that artificially constricts the nature of possibility spaces framing users' potential cognition. Multiple choice tests demand asking clear, static questions about some subject matter where one of n choices is right and n-1 of n choices are wrong. Such a scenario, at an abstract level, is at odds with the nature of modern demands on cognition. Traditional admissions tests focus on product scores (i.e., correctness) not the process of how (i.e., strategy) a user got there. It is vital to understand a user's cognitive process, as cognition by its nature is dynamic across time and tasks.

Beyond content irrelevance, the degree to which today's standardized admissions tests can be "gamed" leads to inequity in opportunity for success. Users who have the resources to master the testing process are more likely to perform better on the assessments. The College Board reported a substantial correlation of r=.42 between socioeconomic status and SAT scores [4]. The SAT's correlation with socioeconomic status is higher than The College Board's self-reported correlation of r=.33 between SAT score and first-year college GPA [4].

Imbellus assessments focus on evaluating how people think instead of what they know. Through our scenarios that take place in our simulation-based environments, we observe details of users' cognitive processes, not just their end choices. We've designed our assessments to discount the high value placed on memory and processing speed in traditional cognitive assessments. The simulation-based assessment discussed in this paper consists of several scenarios embedded in an abstracted natural world environment. Users interact with a series of challenges involving natural terrain, plants, and wildlife (See Figure 1). We designed each scenario as an abstract representation of the problem-solving capabilities and processes required to succeed on the job. This abstraction allows us to transpose skills to a new context with a similar structure to the first—known as far transfer [5]. We strategically chose the natural world as a setting for our tasks because it offers an accessible context for a global population.

Second, our problem-solving assessment focuses on skills mastery rooted in cognitive and learning science theory, as well as an exploration of the nature of work at McKinsey & Company. Together, with McKinsey & Company, we conducted a cognitive task analysis to understand the problem-solving domain [7]. Using this analysis, we developed a problem-solving framework representing seven major constructs (e.g. situational awareness, metacognition, decision-making). We examined on-the-job activities at McKinsey & Company to ensure that the structure of our problem-solving framework was aligned with the practical skills and abilities employees engage in at the firm. This work laid the groundwork for scenario development within our simulation.

Third, our problem-solving assessments focus on the process in which users solve and engage in during the task. We do not just look for correct or incorrect answers; instead, we aim to understand how a user solved a problem and what strategies they engaged in to do so. This novel approach to cognitive testing in the hiring domain provides an abundance of information to better assess which candidates are likely to succeed at the company.

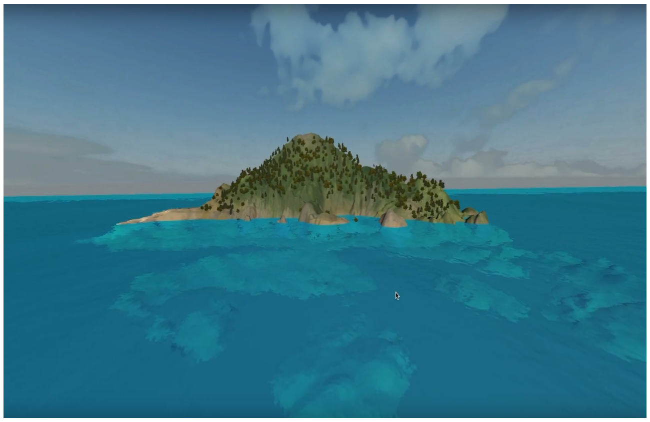


Figure 1. View of natural world simulation environment

We designed each scenario in the assessment based on a set of problem-solving constructs and workplace activities wrapped in a natural world setting. For example, in one scenario, users may be researching and evaluating an infected species in desert terrain. As users play through a scenario, we test them on both their cognitive process and product by capturing their telemetry data. These hovers and clicks are captured as evidence to make inferences about their cognitive processing.

2. OVERVIEW OF SCORE DEVELOPMENT

Imbellus scores were developed using our problem-solving ontology, comprised of approximately 100 constructs, and the cognitive task analysis we conducted with McKinsey & Company. Imbellus scores quantify how users' actions, timestamps, and performance relate to the cognitive constructs within our problem-solving ontology. We derive all Imbellus scores from the users' telemetry data. We then map the scores to one or more problem-solving constructs within our framework.

To create the Imbellus scores, we engaged in a step-by-step process to build, test, and refine each score and its link to the theoretical framework. First, we built expert models for each scenario within our simulation. Expert models help us understand how applicants' cognitive skills manifest in telemetry data. Within our expert models, we outlined the evidence we expected to see in users' behaviors (e.g. efficiency, systematicity) as they complete tasks. We used these evidence statements to develop our Imbellus scores. Following our initial score design, we conducted a series of think-aloud tests aimed at linking specific thinking patterns and behaviors to our scores. We incorporated information from these think-aloud sessions to revise our expert models and scores. We used the initial set of Imbellus scores as a basis for our November 2017 pilot study.

3. PRELIMINARY PILOT OVERVIEW

Using our preliminary Imbellus scores, we conducted a large-scale pilot study in the Fall of 2017. This pilot study tested the predictive capacity of our scores, as well as assessment and simulation environment. We mapped each Imbellus score to one or more of five high level cognitive constructs: critical thinking, decision-making, metacognition, situational awareness and systems thinking. This mapping allows us to build cognitive profiles while also examining the predictive bearing of each score. The pilot study data will be used to inform future designs, validate methodologies, and refine scores.

3.1 Method

Our pilot test, comprised of 527 McKinsey & Company candidates, represented our largest cohort to date. Testing occurred in London, UK from November 13, 2017 through November 17, 2017 and was an optional part of the candidates' interview process with McKinsey & Company. After the conclusion of our game-based assessment, participants completed a survey designed to collect demographic information and user feedback.

Based on survey data, 40% of participants were female, 59% were male, and 1% chose not to provide gender. Based on the Equal Employment Opportunity Commission's guidelines, the ethnic breakdown of the sample was as follows: 52.6% White, 29.7% Asian, 3.9% Hispanic, 4.1% Mixed, 3.3% Black, 2.8% Other, and 3.5% did not specify [2]. Participants' educational backgrounds ranged from humanities-based disciplines to business and engineering. On English proficiency, 56% of the sample reported being a native English speaker, 43% reported being a fluent but non-native English speaker, and 1% reported having a "business-level" proficiency of English.

The participants in our pilot population were given the option of completing our digital assessment after completing the McKinsey & Company Problem-Solving Test (PST), a paper-based assessment. McKinsey & Company administers the PST at proctored test sites. The PST is a traditional cognitive assessment designed to provide insight into applicants' cognitive skills. For the sample of participants who also completed the Imbellus assessment, the proctors told the participants that chose to complete the Imbellus assessment that the outcomes of the assessment would not affect their recruitment process.

Candidates were allotted 60 minutes, the recommended amount of time excluding cases of learner accommodation, to complete the three scenarios. The digital assessment was administered using McKinsey-owned laptop computers in a controlled environment. Along with assessment telemetry and survey data, we collected all scratch paper used by candidates. The assessments took place over the course of 5 days of testing and 29 sessions, none of which experienced significant technical difficulties.

3.2 Creating Construct Profiles

To better understand how participants performed in our assessment, we created a cognitive profile for each participant based on five cognitive constructs: critical thinking, decision-making, metacognition, situational awareness and systems thinking. We already had created theoretical construct affinities for each item score. However, not every item score was predictive. We created a non-negative logistic regression with LASSO regularization to predict the probability a user would pass the first cognitive screen [10].

Before we performed the regression, we imputed missing scores by their median value. All scores were scaled from 0 to 1 using their smallest and largest values. The regression must have non-negative weights because we assume that a higher item score is evidence of higher ability. We used LASSO because of its feature selection properties [11]. The LASSO regularization strength, λ, was found through 10-fold cross-validation. The goal of this step was feature selection, so we chose λ based on a combination of non-zero coefficients and deviance. A λ of 7.68×10^{-3} produced a model with 26 (from 81) non-zero coefficients and a deviance of 1.24 (minimal deviance model = 1.22).

We scaled the resulting item score weights according to their theoretical relevance to each construct. The most relevant scores were multiplied by 3, while relevant scores were multiplied by 2. Marginally relevant scores were not scaled. Item scores that were irrelevant to the construct were set to 0. This created five construct-scaling vectors. The scores for each user were multiplied element-wise by each of the scaling vectors.

These scaled item scores were summed together for each construct. The result was then rescaled by dividing each construct by its highest possible score and transformed into percentile ranks. All construct scores except decision-making had high Pearson correlation (>0.60) with passing McKinsey's multiple-choice Problem-Solving Test (PST). Decision-making had a Pearson correlation of 0.43. The full correlation table between the constructs and passing the PST is displayed below.

Table 1. Correlations between construct scores and PST passing scores

	Meta	ST	SA	DM	CT	PST Pass
Meta	1.00	0.46	0.52	0.43	0.69	0.63**
ST	0.46	1.00	0.70	0.28	0.80	0.67**
SA	0.52	0.70	1.00	0.28	0.79	0.71**
DM	0.43	0.28	0.28	1.00	0.33	0.43**
CT	0.69	0.80	0.79	0.33	1.00	0.65**
PST Pass	0.63	0.67	0.71	0.43	0.65	1.00

All constructs are significantly related to PST pass rate at p<.01

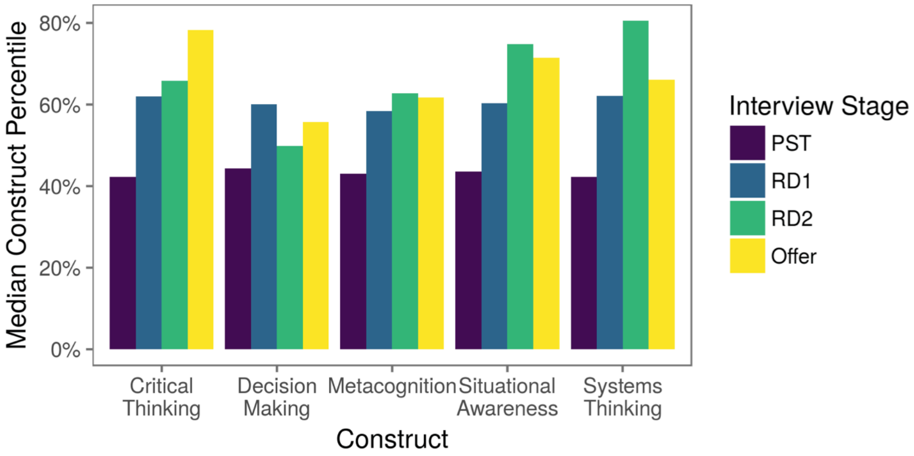


Figure 2. Median Construct Percentile through McKinsey & Company Recruiting Pipeline

The plot above shows the median percentile rank of each of the five construct measures at each stage of the interview process (See Figure 2). Each colored bar in the plot represents the outcome of the interview process. The disposition labeled "PST" signifies that the candidate was screened out before the first interview. "RD1" and "RD2" signify that the applicant did not continue past the first or second round interviews, respectively. "Offer" means that the applicant received an offer from the company.

Below is a table of the median percentile of each of the five constructs at each stage of the interview process along with the median absolute deviation (MAD). This table reveals that preliminary cognitive construct scores are significantly related to success in the interview process. While more work needs to be done to explore this relationship, the initial results are favorable.

Table 2. Median percentile construct score by interview stage.

	PST	RD1	RD2	Offer
Critical Thinking	0.43 (.34)	0.62 (.35)	0.65 (.28)	0.78 (.31)
Decision Making	0.45 (.37)	0.59 (.35)	0.51 (.40)	0.56 (.24)
Metacognition	0.44 (.36)	0.59 (.36)	0.61 (.27)	0.62 (.33)
Situational Awareness	0.44 (.34)	0.6 (.36)	0.74 (.24)	0.71 (.36)
Systems Thinking	0.43 (.35)	0.62 (.35)	0.78 (.28)	0.66 (.19)

Median scores and (Median absolute deviations)

4. CONCLUSIONS & FUTURE WORK

Results from the pilot are promising and show that the Imbellus scores can be used to build out predictive cognitive profiles of candidates. Indeed, these results showed that the cognitive profiles of users were predictive of their success through the McKinsey & Company hiring pipeline. Beyond predictability, these results also show that cognitive processing skills can be captured and quantified using telemetry data within a complex problem-solving task.

To examine the generalizability of these results, we are currently conducting playtests with McKinsey & Company employees and candidates, globally. This extra testing will be used to help us iterate on the design of the assessment and refine our Imbellus scores. In the fall of 2018, we will run a large-scale field test with an expected sample size of over 1000 of McKinsey & Company candidates.

The current version of the simulation is deployed in a secure, proctored environment. In the future, our assessments will be deployed remotely. As such, our assessment will aim to account for performance effects across demographic factors. At its core, Imbellus will leverage a data-driven, artificial intelligence (AI) architecture to prevent cheating. Every user who takes the Imbellus assessment will receive a unique task instance that, on the surface, is varied by its individual properties, complexity, and visual design, while structurally every task version remains consistent in its assessment. Through this approach, Imbellus assessments will prove robust against cheating, hacking, and gaming challenges that face many existing intelligence tests. Our assessments are designed for scale, enabling our team to reach a variety of domains and populations.

Looking beyond this work, we are exploring capabilities beyond problem-solving, including affective skills that are essential for success in the 21st Century workforce. At Imbellus, we aim to provide insightful data points on incoming applicants and current employees that will help companies build successful and sustainable teams in the future.

5. REFERENCES

[1] Assessment & Selection. OPM.GOV. Retrieved from: https://www.opm.gov/policy-data-oversight/assessment-and-selection/other-assessment-methods/cognitive-ability-tests/

[2] Code of Federal Regulations Title 29 – Labor. 1980. Retrieved from: https://www.gpo.gov/fdsys/pkg/CFR-2016-title29-vol4/xml/CFR-2016-title29-vol4-part1606.xml

[3] Gallagher, C. 2003. Reconciling a Tradition of Testing with a New Learning Paradigm. *Educational Psychology Review*. 15, 1, 83-99. DOI=http://www.jstor.org.stanford.idm.oclc.org/stable/2336 1535

[4] Paul R. Sackett, Nathan R. Kuncel, Justin J. Arneson, Sara R. Cooper, & Shonna D. Waters. 2009. Socioeconomic Status

and the Relationship Between the SAT and Freshman GPA: An Analysis of Data from 41 Colleges and Universities. The College Board, New York. Retrieved from: https://research.collegeboard.org/sites/default/files/publications/2012/9/researchreport-2009-1-socioeconomic-status-sat-freshman-gpa-analysis-data.pdf

[5] Perkins, D. N., & Salomon, G. 1992. Transfer of learning. *International encyclopedia of education*. *2*, 6452-6457.

[6] SAT vs. ACT. The Princeton Review. 2018. Retrieved from: https://www.princetonreview.com/college/sat-act

[7] Schraagen, J. M., Chipman, S. F., & Shalin, V. L. (Eds.). 2000. Cognitive task analysis. *Psychology Press.*

[8] The ACT Test for Students. ACT. 2018. Retrieved from: https://www.act.org/content/act/en/products-and-services/the-act.html

[9] Zanderland, L. 1998. Measuring Minds. *Cambridge University Press*, Cambridge, UK.

[10] Mhukrishnan, R., and R. Rohini. "LASSO: A Feature Selection Technique in Predictive Modeling for Machine Learning." In *2016 IEEE International Conference on Advances in Computer Applications (ICACA)*, 18–20, 2016. https://doi.org/10.1109/ICACA.2016.7887916.

[11] Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, *33*(1), 1–22.

Forgetting curves and testing effect in an adaptive learning and assessment system

Jeffrey Matayoshi
McGraw-Hill Education/
ALEKS Corporation
Irvine, CA
jeffrey.matayoshi@aleks.com

Umberto Granziol
University of Padova
Padova, Italy
umberto.granziol@phd.unipd.it

Christopher Doble
McGraw-Hill Education/
ALEKS Corporation
Irvine, CA
christopher.doble@aleks.com

Hasan Uzun
McGraw-Hill Education/
ALEKS Corporation
Irvine, CA
hasan.uzun@aleks.com

Eric Cosyn
McGraw-Hill Education/
ALEKS Corporation
Irvine, CA
eric.cosyn@aleks.com

ABSTRACT

In the context of an adaptive learning and assessment system, ALEKS, we examine aspects of forgetting and aspects of a 'testing effect' (in which the act of simply being presented a problem in an assessment seems to assist in the learning process). Using a dataset consisting of over six million ALEKS assessments, we first look at the trend of student responses over the course of the assessment, finding little evidence for such a testing effect. We then refine our approach by looking at cases in which a question is repeated in an assessment; repeats are possible because some question is always chosen at random in an assessment for data-collection purposes. We find evidence of a testing effect for higher-performing students; for lower-performing students, we find a decreased willingness to attempt an answer the second time a problem is presented. Then, turning to forgetting, we find that the content representing the "high points" of a student's learning sees a more precipitous drop in the student's memory than does other content (perhaps because the "high point" skills and concepts may not have been practiced or developed much since the original learning event). Consequences and possible improvements for the ALEKS system, and also a brief comparison to recent work in the modeling of forgetting, are mentioned.

Keywords

Knowledge space theory, adaptive learning, forgetting curves, testing effect

1. INTRODUCTION

ALEKS, which stands for "**A**ssessment and **LE**arning in **K**nowledge **S**paces", is a web-based, artificially intelligent, adaptive learning and assessment system [13]. The artificial intelligence of ALEKS is a practical implementation of knowledge space theory (KST) [5, 7, 8], a mathematical theory that employs combinatorial structures to model the knowledge of learners in various academic fields of study including math [11, 15], chemistry [9, 18] and even dance education [19].

2. BACKGROUND

Memory and forgetting is an area that has seen significant research, pioneered by the late-nineteenth century work of Ebbinghaus with his 'forgetting curves' [2, 6]. Ebbinghaus posited that memory, as measured, say, by the ability to recall words presented in a list, decays exponentially with time; one such exponential model is given in Equations (7.1) and (7.2) in Section 7 below. A great deal of study has been done on the possible effects of various experimental conditions, such as whether the experiment probes explicit or implicit memory [12], the effect of the physical context in which the learning and recall take place [3, 17], and the extent to which the content is meaningful for the participant [10, 14], among many other experimental conditions. In the current paper, we will examine forgetting in the context of the adaptive learning and assessment system ALEKS, attempting to isolate the effect of aspects of the adaptivity on forgetting.

We will also look at a kind of 'testing effect' in which the act of simply being presented content in the adaptive assessment seems to assist in the learning process [1, 4]. We use the term 'testing effect' somewhat loosely here, as our use differs from that typically seen in the literature, since, for example, our situation does not include systematic feedback [16]. We use the term only to refer to a situation in which recall (or skill, or confidence) seems improved as content is encountered during an assessment.

In KST, an *item* is a problem that covers a discrete skill or concept. Each item is composed of many examples called *instances*; these instances are carefully chosen to be equal in difficulty and to cover the same content. A *knowledge state* in KST is a collection of items that, conceivably, a student at any one time could know how to do. In other words, roughly speaking, a set of items is a knowledge state if some student could know how to do all of the items in the set and not know how to do any of the items outside the set. For example, the empty set and full set are always considered knowledge states.

Another important concept from KST is the *inner fringe* of a knowledge state. An item is contained in the inner fringe of a knowledge state when the item can be removed from the state and the remaining set of items forms another knowledge state. Intuitively, the inner fringe items are the "high points" of a student's knowledge, as they are not prerequisties required to master any of the other items in the knowledge state. This concept will be important for our work on forgetting in Sections 5 and 6.

While using the ALEKS software, the student is guided through a course via a cycle of learning and assessments. Each assessment (described below) updates the system's assignment of a knowledge state to the student. Then, in the learning mode, the student is given problems to practice based on her knowledge state, with the system tracking the student's performance and continually updating the student's knowledge state. Subsequent assessments then modify the knowledge state as needed, and the process continues.

Each ALEKS assessment has about 15 to 29 questions, with each question comprising the presentation of some item to the student. The item is chosen in an adaptive way, that is, chosen based on the student's previous responses during the assessment. (More specifically, the item is chosen to be maximally informative for the system's evaluation of the student. The effect is that the assessment adapts to the level of the student, not necessarily becoming easier or harder for the student, as the assessment continues.) The student can elect to give an answer for the item, in which case her response is classified by the system as correct or incorrect, or she can choose to respond "I don't know," which she is encouraged to do if she has no idea how to approach the item. In addition, in each assessment, an *extra problem* is chosen uniformly at random from all of the items in the course and presented to the student as a question in the assessment. The student's response to the extra problem does not affect the system's evaluation of the student.

3. EXTRA PROBLEM BY RANK

For our first analysis, we will look at how responses (correct, incorrect, or "I don't know") to the extra problem evolve during the assessment. In other words, does the question rank of the extra problem have an effect on students' responses? (By *question rank*, we mean the point in the assessment at which the question is asked, that is, the question number.) One hypothesis is that the extra problem success rate would increase throughout the assessment. (By *success rate*, we mean the proportion of the responses that are correct.) For example, it is possible that simply by working through repeated assessment questions, students experience a boost in performance; we will consider this phenomenon as a type of 'testing effect' [1, 4, 16]. One could imagine that this effect would be more pronounced after a long academic break, such as a summer or winter vacation, since the skills required for a particular course could suffer from a lack of recent use, and being assessed on these skills could help to sharpen them. As another example, there could be user interface issues for a student who is unfamiliar with the ALEKS system. Since the large majority of ALEKS problems require open-ended solutions, rather than multiple choice responses, it is possible that students would improve in performance as they became accustomed to the ALEKS interface. In both of these scenarios, the effect, if it existed, would seem to be more apparent earlier in a course, so we will look at data from ALEKS *initial assessments*, which are the assessments given at the start of an ALEKS course.

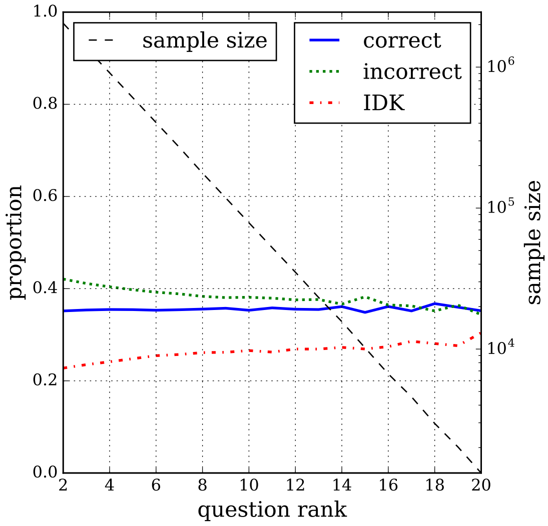


Figure 1: Proportions of responses to the extra problem by question rank for initial assessments. The types of responses are correct, incorrect, and "IDK" ("I don't know"). Note that the sample size is shown on a logarithmic scale.

Note that both of the hypothesized effects in the previous paragraph would result in an increased extra problem success rate as the assessment progresses. However, one effect that would possibly lower the success rate, and that has been observed anecdotally, is that of assessment fatigue: as an assessment goes on, students may be more likely to respond incorrectly or not at all. This effect may be amplified by the open-ended answer interface used by ALEKS, which could make it more appealing for a student to respond "I don't know" rather than make the effort to input a complete answer.

To start, we will look at a dataset consisting of 6,132,681 initial assessments, grouping the responses to the extra problem by question rank. The results can be seen in Figure 1. The first thing to note is that the success rate (the proportion of correct responses) does not increase as the assessment goes on; its curve is essentially flat. Thus, whatever testing effect there may be is overwhelmed by other factors. In particular, the rate at which students answer "I don't know" shows a steady rise as the question rank increases, and the incorrect rate shows a corresponding decrease; keeping in mind that the extra problem is a randomly chosen problem that is asked at a randomly chosen point in the assessment, we see evidence that students are experiencing some sort of fatigue. As students get further along in the assessment, they seem less willing to attempt a problem and more will-

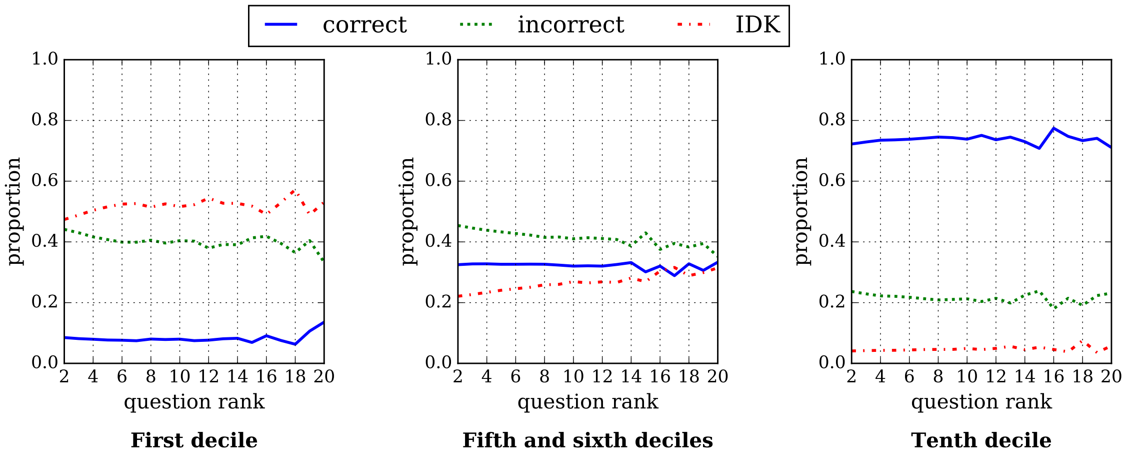


Figure 2: Proportions of responses to the extra problem by question rank for initial assessments, with percentage scores in (i) the first decile, (ii) the fifth or sixth decile, and (iii) the tenth decile.

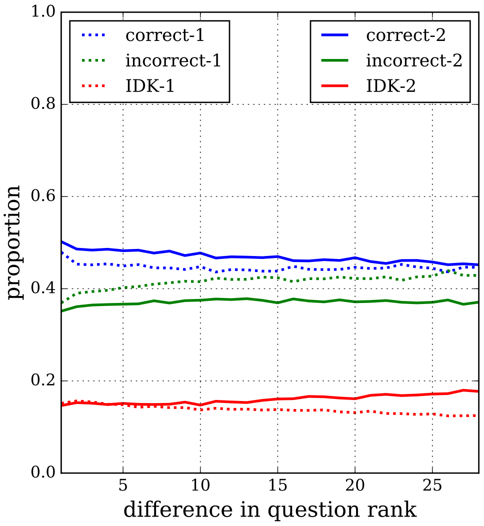


Figure 3: Proportions of responses for repeated items in initial assessments. The horizontal axis gives the difference in question rank between the two occurrences. The dotted curves (e.g., "correct-1") give the response proportions for the first occurrence, and the solid curves (e.g., "correct-2") give the response proportions for the second occurrence. The top set (pair) of lines represents the correct responses, the middle set represents the incorrect responses, and the bottom set represents the "I don't know" responses.

ing simply to respond "I don't know." What is striking is that, since the proportion of correct responses holds steady, it appears that many of these "I don't know" responses would have been incorrect responses earlier in the assessment; thus, one can alternatively interpret this as students being more "accurate" or "honest" in their self-assessment of the items they are capable of answering correctly.

To better understand these observed effects, we look more closely at the data based on the results of the initial assessment. We define the student's *initial assessment score* to be the percentage of the items in the course that are in the student's knowledge state according to the initial assessment, which gives a measure of the student's knowledge at the start of the course. Figure 2 shows the same results as in Figure 1, but this time separately for the three groups of students with initial assessment scores in (i) the first decile of all of the scores in the dataset, (ii) the fifth or sixth decile, and (iii) the tenth decile. From the plots in Figure 2, we can see that the (putative) fatigue effect is dependent on the group. The students in the middle group, with scores in the fifth and sixth deciles, seem to be most heavily affected, with a large increase in the "I don't know" rate as the assessment progresses. On the other hand, the students in the tenth decile show hardly any change over the course of the assessment, with the rates being mostly constant. Lastly, the students in the first decile are somewhere in the middle, with a sharp increase in the "I don't know" rate for the first few questions, and then a relatively flat curve thereafter.

4. REPEATED QUESTION

In the previous section, we saw that over the length of an assessment, the success rate was relatively flat. Thus, if there is any sort of boost from a testing effect, it is overwhelmed by other factors and is not apparent in our initial analysis. In the current section, we will take a more targeted approach and look at cases in which an item appears multiple times in an assessment. In particular, we will look at cases in which an item is first asked as an extra problem and then asked later in the same assessment as a "regular" question. (It is important to note that a different instance of the item is given each time, so that even though the type of problem being tested is the same, the particular example being presented is different.) Using a dataset composed of 644,462 initial assessments, each having some item repeated during the assessment, we can compare the success rates for the two occurrences of the repeated item. The results of this analysis are shown in Figure 3, where the horizontal axis gives the difference in question rank between the two occurrences. We can see that, overall, there is a gap between the success rates for the first and second occurrences, with the students being more successful on the second attempt. However, as with the analysis in Section 3, grouping the students by their initial assessment scores shows some pronounced differences. Figure 4 shows the results for students with initial assessment scores in the first decile; here,

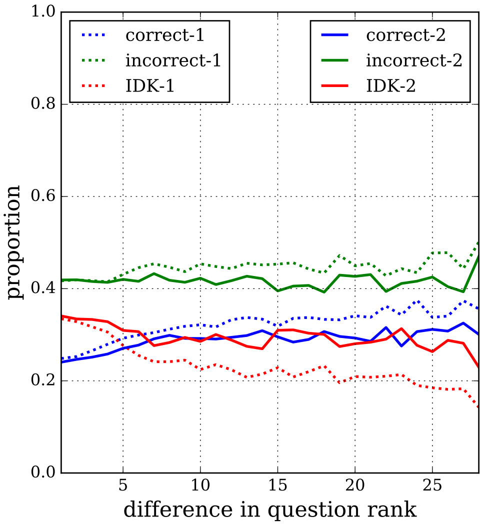


Figure 4: Proportions of responses for repeated items in initial assessments, for students with a percentage score in the first decile. The horizontal axis gives the difference in question rank between the two occurrences. Using the ordering at the leftmost edge of the horizontal axis, the top set (pair) of lines represents the incorrect responses, the middle set represents the "I don't know" responses, and the bottom set represents the correct responses.

in contrast to the overall trend, the students do worse on the second attempt. Interestingly, both the correct and incorrect rates *decrease* on the second attempt, with the "I don't know" rate showing a correspondingly large increase. Thus, it seems that the overall trend for students in this category is to be less confident, or at least less willing to attempt an answer, on their second attempt at a repeated item.

On the other hand, Figure 5 shows a much different trend for the students in the tenth decile. The "I don't know" rate is unchanged from the first attempt to the second, while a significant portion of the incorrect responses from the first attempt seemingly become correct responses in the second attempt. Thus, for students whose initial assessment scores are at the high end, it does appear that having multiple attempts at a problem gives a significant advantage.

As described in the previous section, the majority of students taking an initial assessment are returning from a break in schooling, often due to summer vacation. Thus, taking an ALEKS initial assessment may be one of the first chances in several months for a student to practice her math skills; in such a case, the simple act of working on an item may help the student recall some of the needed skills, or even to figure out new skills, which may then translate to greater success on a subsequent appearance of the item.

5. INNER FRINGE FORGETTING CURVE

In the next two sections we will examine forgetting as it applies to the ALEKS system. We will begin by looking at how the success rate of an inner fringe item changes as a function of the time since the item was first learned (with "learning" an item amounting to demonstrating a certain amount of success on the item in the learning mode). To do this, we will use data gathered from 286,345 ALEKS *progress assessments*, which are assessments given to a student after he has spent some time in the learning mode. The purpose of a progress assessment is to verify the student's recent learning. The progress assessments we examine here are limited to those for which the item presented as question 1 of the assessment is contained in the inner fringe of the student's knowledge state. Since the assessment is adaptive, we restrict our analysis to the first item presented to avoid any bias from the item-selection algorithm. We also look only at inner fringe items to reduce any bias that may come from the student working on items with related content: As mentioned, items in the inner fringe of a student's knowledge state are not required to master any of the other items in the knowledge state, so if an item is in the inner fringe, the student has not spent time learning new concepts that build on that specific item. For each of these progress assessments in which question 1 is an item appearing in the inner fringe of the student's knowledge state, we compute the number of days from the time the student learned the item to the time the item appeared in the progress assessment.

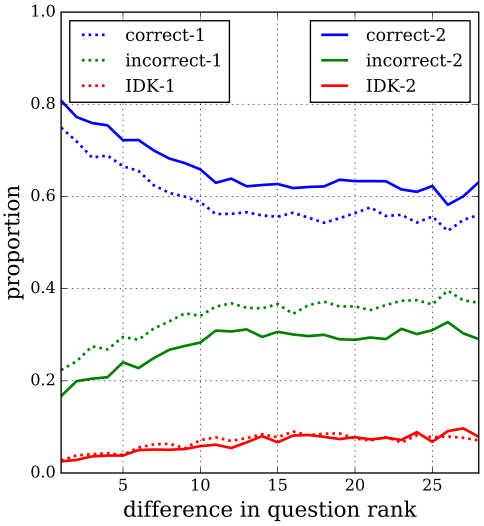


Figure 5: Proportions of responses for repeated items in initial assessments, for students with a percentage score in the tenth decile. The horizontal axis gives the difference in question rank between the two occurrences. The top set (pair) of lines represents the correct responses, the middle set the incorrect responses, and the bottom set the "I don't know" responses.

The results are in Figure 6. In this figure, the solid curve (the one near the top of the figure) can be considered a forgetting curve [2, 6]. As shown, there is a clear decrease in the success rate as the number of days since the item was learned increases, while the rates of incorrect and "I don't know" responses both increase. The changes are greatest over the initial few days and then flatten out somewhere between one and two weeks. As an aside, we can also see in Figure 6 the weekly cycle of student use, which causes the sample size to peak every seven days.

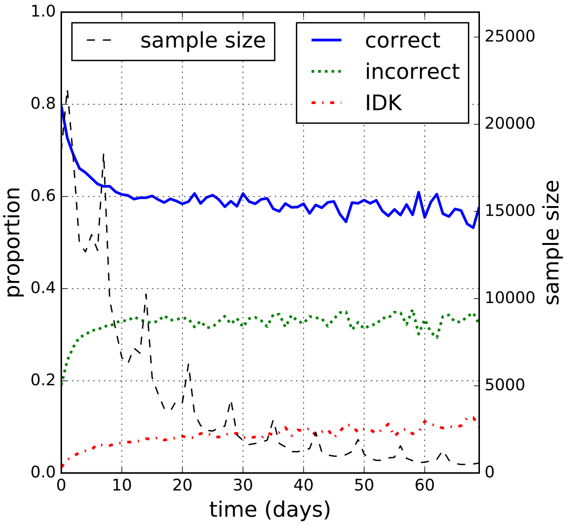


Figure 6: Proportions of responses as a function of the time (in days) since the item appearing as question 1 in a progress assessment was learned.

6. EXTRA PROBLEM OVER TIME

For the next part of our analysis, we again use data generated by ALEKS progress assessments. Rather than looking at the first item presented, however, we instead focus on the extra problem. Restricting our analysis to extra problems that have been previously learned by the student, we can again look at the response rates as a function of the time since the item was first learned. Using data from 72,045 progress assessments that fit the criteria, we show the results in Figure 7. (Furthermore, for ease of comparison, we display the information from Figures 6 and 7 in Figure 8.)

While there is a drop in the success rate over the first few days, in comparison to Figure 6 this drop is less pronounced, and it levels off within a shorter amount of time. The reason for this is most likely that we are no longer looking only at items in the inner fringe of the student's knowledge state. Recall that, if an item is in the student's inner fringe, then the student has not (at least in theory) mastered any subsequent material that requires complete mastery of that item. However, this no longer holds for a randomly chosen item from the student's knowledge state; for example, the student may have mastered one or more subsequent items that require complete mastery of the extra problem, which may have the effect of reinforcing the learning of the concepts in the extra problem. Thus, the flatter nature of the extra problem forgetting curve can be viewed as a consequence of the adaptive nature of the ALEKS system, which serves to reinforce the original learning.

On the other hand, the success rate on the extra problem does exhibit a noticeable decline over the first several days after the item is learned. It is during this period that more targeted review and/or practice may be beneficial.

7. DISCUSSION AND FUTURE WORK

In the above analyses, we observed the following: (1) Students, especially those near the middle of the range in content knowledge, tend to replace incorrect responses with ones of "I don't know" as the assessment progresses; (2) Students at the upper end in content knowledge tend to improve on an item the second time the item is asked in an assessment, while students on the lower end tend to do worse the second time, or at least to become less confident; (3) Items that give the "high points" of a student's learning see a more precipitous drop in the student's memory than do other items (perhaps because the skills and concepts in these "high point" items may not have been practiced or developed much since the original learning event). A possible improvement to the ALEKS learning and assessment software based on these observations may be to introduce pointed feedback during an assessment to provide encouragement or guidance to students who are at risk of fatiguing or declining in confidence. Another may be to have a dedicated review period for "high point" items, perhaps given in conjunction with a progress assessment itself, to help with immediate forgetting.

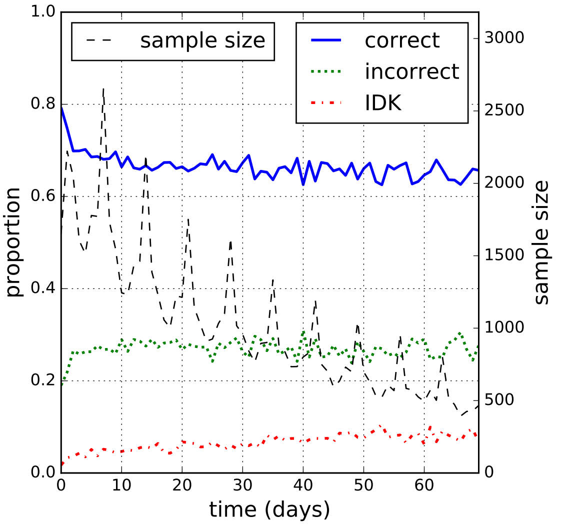


Figure 7: Proportions of responses as a function of the time (in days) since the extra problem appearing in a progress assessment was learned.

In addition to suggesting improvements to the ALEKS system, our analyses may both inform and be informed by the extensive literature on memory. Take, for example, the particular forgetting curve analysis in [2], in which the authors examine models of forgetting given by

$$R(t) = a + (1 - a) \times b \times P(t), \quad 0 < a, b < 1, \tag{7.1}$$

for different functions $P(t)$. Here, $R(t)$ gives the probability of retention at time t, and a and b are parameters. One such function $P(t)$ examined in [2] is

$$P(t) = (1 + t)^{-\beta}, \tag{7.2}$$

in which $\beta > 0$ is a parameter. Fitting $R(t)$ (with this form of $P(t)$) to the success rates for question 1 and extra problem data gives the smooth curves shown in Figure 8. The fit is strong, with the $R(t)$ curves closely following the trend of the data. (The increasing jaggedness of the correct curves in Figure 8 stems from the decreasing sample sizes, as shown in Figures 6 and 7.) For reference, we report that for this fit, the parameters a, b and β are estimated to be 0.55, 0.56

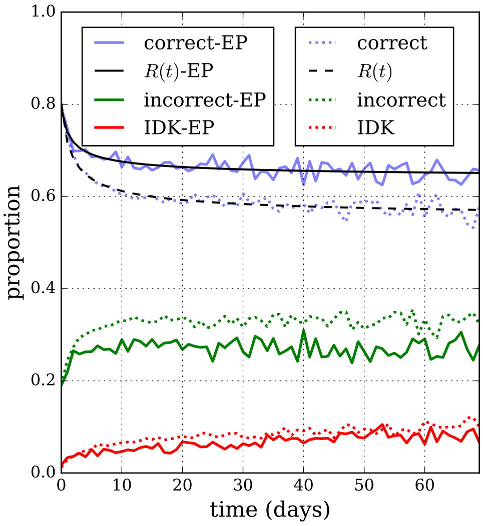


Figure 8: A direct comparison of Figures 6 and 7. The solid curves (e.g., "correct-EP") are from Figure 7, giving the proportions of responses as a function of the time (in days) since the extra problem was learned. The dotted curves are from Figure 6, giving the proportions of responses as a function of the time (in days) since the item appearing as question 1 was learned. Also shown are the curves obtained from fitting $R(t)$ given by Equation (7.1) (with $P(t)$ as in (7.2)) to the data.

and 0.59, respectively, for the question 1 curve; for the extra problem curve, these parameters are estimated to be 0.64, 0.42 and 0.58, respectively.

It is a natural next step to implement such a model to improve students' experiences using ALEKS by improving, for example, the scheduling of progress assessments, the item-selection algorithm, and the timing and content of review periods for newly learned items.

Further, it is feasible that the very large data sets examined in this paper may contribute to the discussion of competing mathematical models of forgetting. For example, the authors in [2] also examine forgetting functions of the form

$$R(t) = a + (1 - a) \times be^{-\alpha t} \tag{7.3}$$

and of the form

$$R(t) = 0.116 + (1 - 0.116) \times b \times (1 + \gamma t)^{-\beta}, \tag{7.4}$$

comparing the various special cases of (7.1) given by (7.2)-(7.4). Our data would likely contribute to this and similar discussions.

8. REFERENCES

[1] Agarwal, P., Bain, P., and Chamberlain, R. The value of applied research: Retrieval practice improves classroom learning and recommendations from a teacher, a principal, and a scientist. *Educational Psychology Review 24* (2012), 437–448.

[2] Averell, L., and Heathcote, A. The form of the forgetting curve and the fate of memories. *Journal of Mathematical Psychology 55* (2011), 25–35.

[3] Baddeley, A., Eysenck, M. W., and Anderson, M. C. *Memory.* Psychology Press, New York, 2009.

[4] Carrier, M., and Pashler, H. The influence of retrieval on retention. *Memory and Cognition 20* (1992), 632–642.

[5] Doignon, J.-P., and Falmagne, J.-C. Spaces for the assessment of knowledge. *International Journal of Man-Machine Studies 23* (1985), 175–196.

[6] Ebbinghaus, H. *Memory: A Contribution to Experimental Psychology.* Originally published by Teachers College, Columbia University, New York, 1885; translated by Henry A. Ruger and Clara E. Bussenius (1913).

[7] Falmagne, J.-C., Albert, D., Doble, C., Eppstein, D., and Hu, X., Eds. *Knowledge Spaces: Applications in Education.* Springer-Verlag, Heidelberg, 2013.

[8] Falmagne, J.-C., and Doignon, J.-P. *Learning Spaces.* Springer-Verlag, Heidelberg, 2011.

[9] Grayce, C. A commercial implementation of knowledge space theory in college general chemistry. In *Knowledge Spaces: Applications in Education*, J.-C. Falmagne, D. Albert, C. Doble, D. Eppstein, and X. Hu, Eds. Springer-Verlag, 2013, ch. 5, pp. 93–114.

[10] Hanley-Dunn, P., and McIntosh, J. L. Meaningfulness and recall of names by young and old adults. *Journal of Gerontology 39* (1984), 583–585.

[11] Huang, X., Craig, S., Xie, J., Graesser, A., and Hu, X. Intelligent tutoring systems work as a math gap reducer in 6th grade after-school program. *Learning and Individual Differences 47* (2016), 258–265.

[12] McBride, D. M., and Dosher, B. A. A comparison of forgetting in an implicit and explicit memory task. *Journal of Experimental Psychology: General 126* (1997), 371–392.

[13] McGraw-Hill Education/ALEKS Corporation. What is ALEKS? `https://www.aleks.com/about_aleks`.

[14] Paivio, A., and Smythe, P. C. Word imagery, frequency, and meaningfulness in short-term memory. *Psychonomic Science 22* (1971), 333–335.

[15] Reddy, A., and Harper, M. Mathematics placement at the University of Illinois. *PRIMUS 23* (2013), 683–702.

[16] Roediger, H. L., and Butler, A. C. The critical role of retrieval practice in long-term retention. *Trends in Cognitive Sciences 15* (2011), 20–27.

[17] Smith, S. M. Remembering in and out of context. *Journal of Experimental Psychology: Human Learning and Memory 4* (1979), 460–471.

[18] Taagepera, M., and Arasasingham, R. Using knowledge space theory to assess student understanding of chemistry. In *Knowledge Spaces: Applications in Education*, J.-C. Falmagne, D. Albert, C. Doble, D. Eppstein, and X. Hu, Eds. Springer-Verlag, 2013, ch. 6, pp. 115–128.

[19] Yang, Y., Leung, H., Yue, L., and Deng, L. Automatic dance lesson generation. *IEEE Transactions on Learning Technologies 5* (2012), 191–198.

LeCoRe: A Framework for Modeling Learner's preference

Kumar Abhinav
Accenture Labs
Bangalore, India
k.a.abhinav
@accenture.com

Venkatesh Subramanian
Accenture Labs
Bangalore, India
venkatesh.subramania
@accenture.com

Alpana Dubey
Accenture Labs
Bangalore, India
alpana.a.dubey
@accenture.com

Padmaraj Bhat
Accenture Labs
Bangalore, India
padmaraj.bhat
@accenture.com

Aditya D. Venkat
Accenture Labs
Bangalore, India
d.venkat.aditya
@accenture.com

ABSTRACT

Learning & development (L&D) is an important strategic factor for sustainable business growth of any organization. L&D has become an integral part of an organization, considering the fast-paced growth of industries. Success of learning and development program depends on how well it identifies the critical skill gaps in its workforce and bridges those gaps by considering individual learner's strengths and aspirations while recommending learning opportunities. Such recommendation requires a rich information about learners and learning opportunities. In this paper, we propose a framework that recommends learning opportunities to learners based on their preferences. We also propose a way to connect users of similar interests on the platform to improve course engagement. We developed a conversational AI agent that assist the learner in their journey. We evaluated our approach on the dataset consisting of 5,000 learners, and 49,202 courses. Our approach performed significantly better than the baseline approach.

Keywords

Learning & Development, Recommendation, Personalization

1. INTRODUCTION

Learning & Development (L&D) program plays a vital role in the overall talent management of any organization. The primary functions of the L&D program are to - 1. Identify the skills that are needed to achieve business goals, 2. Understand the skill gaps in its workforce, 3. Define a plan to close the gaps, and 4. Successfully deploy that plan. These are the crucial steps to create a strong pipeline of employees with appropriate skills required for current and future business needs. It also helps in understanding employees' learning needs based on their career aspiration and provides them a personalized learning plan. To provide a personalized learning plan, one needs to have access to variety of information such as employees' data, their goals, their position, and preferences. Along with this, one also needs to have an understanding of business requirements, courses available internally and externally, etc. With changing times, large employee base, global footprint, and varied profiles to train, monitoring and processing of data has become difficult. Large amount of learning content is available within the organization along with various other external sources of learning such as Massive Open Online Course (MOOC). Most of the content is not personalized and so the learners spend an inordinate amount of time in identifying the relevant content to leverage. We have used employee and learner interchangeably. A number of studies focus on recommending courses to learners [2] [6]. However, these existing recommendation systems have three major shortcomings - 1. They work on a limited set of information and do not utilize the rich set of information available about learners and courses. These information are dispersed at several enterprise systems making it difficult to consume. 2. They do not consider the scenario of a new learner whose course preference data is not available. 3. They do not utilize the social connections that learners may have for enhanced peer learning with learners of similar interests.

To address these challenges, we propose a recommendation framework that models the learner's behavior from the data available about them on different platforms. The framework is a part of the system "LeCoRe" that helps the learners in selecting the right learning content based on their history. The learners' preference model is built from the courses that learners have registered or completed in the past and their profile information captured through several internal as well as external platforms such as LinkedIn, Accenture People [1] etc.
The main contributions of the paper are as follows:

1. An ensemble based learning content recommender: We propose an ensemble based approach of content and collaborative filtering to evaluate the recommendation

[1] Accenture Internal portal to manage employee's details

framework. The results show a significant improvement over the baseline approach.

2. Conversation AI agent: We propose a conversational AI agent that assist the learner in their journey on the learning platform.
3. A system to connect learners with similar interests: The system also promotes an effective engagement among learners by establishing a connection with other similar learners within the platform.

The proposed system brings three main advantages. Firstly, the system improves the ease with which the learners select the learning content, matching with their interest. Secondly, the right content helps the learners in their career growth. Thirdly, the analytic techniques we have devised make use of much richer and contextual data available about learners. The remainder of this paper is structured as follows: Section 2 discusses the related work study in course recommendation. In subsequent section 3, we discuss the recommendation framework. In Section 4, we discuss the system architecture. We describe our dataset in Section 5 and discuss evaluation methodology and results in Section 6. Section 7 discusses the implementation as a tool. Finally, Section 8 concludes with summary of our findings.

2. RELATED WORK

Several studies have been performed in the area of recommending courses to learners. Aher et al. [6] combined clustering and association rule mining algorithms to recommend courses using historical data. Apaza et al. [3] proposed the course recommendation system based on the topic modeling technique. They computed the semantic similarity between the topics extracted from college course syllabus with the topics of MOOCs based courses and then applied content based approach to recommend relevant online courses to college students. However, it didn't consider the learner's history and lacked personalization. Fatiha et al. [7] applied Case Based Reasoning (CBR) based approach to find courses for learners that best fit their personal interests. They used Levenshtein distance to measure the similarity between the cases' attributes. Piao et al. [1] compared the three user modeling strategies based on job title, education and skills available on user's LinkedIn profiles, for personalized MOOC recommendations. They applied dot product similarity between user and course profiles, and then ranked the user's courses based on the similarity. Jiye et al. [5] conducted an experiment on edX platform to identify the factors that contribute to student enagagement in MOOC discussion forums. Jiezhong et al. [2] analyzed the key factors that influence users' engagement in MOOCs using the data collected from xuetangX, one of the largest MOOCs platform from China. Our approach can be differentiated with the state-of-the-art approaches along three dimensions. Firstly, we are the first to apply Deep Learning based approach to model learner's preference and recommending learning content. The system also addresses the problem of new learners with no learning history. Secondly, our platform provides bot assisted learner journey and also helps the learner to connect to social community to promote interaction among learners. Thirdly, our approach is much more comprehensive and models learner's preference over various dimensions of learner and course profiles.

3. FRAMEWORK

We propose a recommendation framework that models the learner's preference. We apply an ensemble based approach where we combine the predictions of Collaborative filtering and Content based techniques.

1. **Collaborative-based approach**: It is one of the most popular and powerful techniques used in recommendation systems. Collaborative Filtering approach (CF) builds the user's interest by collecting preferences of many other users [22]. We employ three popular collaborative filtering techniques:
   (a) **Singuar Value Decomposition**: It is one of the most popular Matrix factorization based techniques that involves decomposing a sparse user-item matrix into two low rank latent matrices that represents user factors and item factors. The missing ratings are then predicted from the inner product of these two factor matrices [22].
   (b) **Slope-One**: The Slope-One approach considers information from other users who rated the same item and from the other items rated by the same user [4].
   (c) **K-Nearest Neighbor**: K-Nearest Neighbor approach takes into account either the items or the users that are similar. This is captured using similarity metrics like Pearson Correlation, Euclidean Distance etc. It predicts rating of the item given by the user based on the weighted average of top-k similar users.
2. **Content-based approach**: The collaborative based approach considers the rating given by learners for different courses. However, it doesn't consider the learner's profile and the content of the courses. We employ content based approach that considers the personal characteristics of the learner and course information registered or completed by the learner. Personal characteristics of the learner includes Skills (skillset of the learner), Geography (geographical unit of the learner), Experience (years of experience) and Industry. Course information includes the course title, course description, course content type (such as web-based etc.). The title and description of the course is represented as topics vector using Topic Modeling techniques. Topic modeling [17] techniques are probabilistic model that have been used to identify topics within the text documents. Latent Dirichlet Allocation (LDA) [16], one of the popular topic modeling techniques, extracts topic information from unstructured text as probability distribution of words. LDA model is used as feature descriptor for course title, course description and profile description of the learner. We pose it as a regression problem. The algorithm predicts the learner's rating to a course using learner's profile and course information as the features. We apply Deep Neural Network based approach to predict the rating for the course. We use Multi-layer Perceptron [9] (also feed-forward neural network) model that consists of input layer, 6 hidden layers and an output layer. Multi-layer Perceptron is a supervised algorithm that learns

a non-linear function for classification or regression. It utilizes a backpropagation technique to optimize the weights so that the neural network can learn to map arbitrary inputs to outputs during training [15]. The predicted output of the network is compared to the expected output and an error is calculated. The error is then back propagated through the network, one layer at a time, and the weights are updated according to the amount contributed to the error. We use Dropout regularization technique to prevent neural networks from overfitting [8]. This is a technique where randomly selected neurons within the network are ignored while training the model. The dropout is applied after each hidden layers. The "Relu" activation function is applied to all the hidden layers. Activation functions convert an input signal of node to an output signal and introduce non-linear properties to neural network.

Many times, the system does not have much information about the new learner's preferences in order to make recommendations. This scenario is referred as "Cold Start", which is a classical problem in recommendation system. In order to build the profile of the new learner, we applied the concept of transfer learning where we identified the learners who are similar to the new learner and used their preferences. The concept of similar learner also helps in matching learners who are mutually interested, and likely to communicate with each other based on their profile characteristics and course enrollment. One of the main reasons for very high dropouts rate in MOOC is lack of engagement among the users [20]. Studies [18][21] have shown that collaboration among the learners promotes better engagement and reduces dropouts on MOOC platform . This would help in fostering the communication between the learners and forming social community of learners. We used the following similarity measures to compute the similarity between learners.

1. Similarity between the projects completed by the learners. We applied content matching approach "Latent Dirichlet Allocation" to find the similarity between their projects.
2. Similarity between their profile characteristics such as Profile Overview and skills
3. Similarity between the description of the courses that the learners have enrolled.

The steps for computing similarity between the learners is described in Algorithm 1. The algorithm computes the similarity between the learners i.e., the distance between a learner with every other learners based on the learner's history. This is computed offline and updated at certain intervals. We applied user-based Nearest Neighbor (K=5) approach to find the similar learners.

4. SYSTEM ARCHITECTURE

The framework of our approach named LeCoRe, shown in Figure 1. Our recommendation approach combines both content-based and collaborative filtering techniques. The proposed recommendation approach consists of four major phases:

Algorithm 1 Learner-Learner Similarity

```
Input: Learner's profile information
Output: Matrix representing the similarity score between learners
Initialize all the diagonal elements of matrix to 1 and rest 0
for $i \in \{1, ..., N\}$ do
    for $j \in \{i+1, ..., N\}$ do
        Apply LDA on description of projects completed by learners $l_i$ and $l_j$
        Compute the cosine similarity between Project Description Topics vector of $l_i$ and $l_j$
            $cos\_sim(PD_{l_i}, PD_{l_j}) = \frac{\vec{PD}_{l_i}.\vec{PD}_{l_j}}{||PD_{l_i}||.||PD_{l_j}||}$
        Apply LDA on profile overview of learners $l_i$ and $l_j$
        Compute the cosine similarity between Profile overview Topics vector of $l_i$ and $l_j$ as:
            $cos\_sim(PO_{l_i}, PO_{l_j}) = \frac{\vec{PO}_{l_i}.\vec{PO}_{l_j}}{||PO_{l_i}||.||PO_{l_j}||}$
        Calculate the skill/concepts similarity between learners $l_i$ and $l_j$ as:
            $Skill\_similarity(S_{l_i}, S_{l_j}) = \frac{|S_{l_i} \cap S_{l_j}|}{|S_{l_i} \cup S_{l_j}|}$
        where $S_{l_i}$ and $S_{l_j}$ is the set of skills possessed by learners $l_i$ and $l_j$ respectively.
        Apply LDA on description of courses enrolled by learners $l_i$ and $l_j$
        Compute the cosine similarity between Course Description Topics vector of $l_i$ and $l_j$ as:
            $cos\_sim(CD_{l_i}, CD_{l_j}) = \frac{\vec{CD}_{l_i}.\vec{CD}_{l_j}}{||CD_{l_i}||.||CD_{l_j}||}$
    end for
    Calculate learner-learner similarity score as:
        $L_{ij} = (cos\_sim(PD_{l_i}, PD_{l_j}) + cos\_sim(PO_{l_i}, PO_{l_j})$
        $+ Skill\_similarity(S_{l_i}, S_{l_j}) + cos\_sim(PO_{l_i}, PO_{l_j}))/4$
end for
```

1. Learners Similarity: The system retrieves the learner's profile information which consists of individual characteristics of the learner such as profile overview, projects, etc. as well the learner's course history. The system utilizes the learners' profile to compute the similarity among the learner, as discussed in Algorithm 1. The output will be learner-learner similarity matrix which will have the similarity score of one learner with rest of the learners. The learner-learner similarity matrix will be stored in the database. These computations are performed offline and updated after certain intervals.

2. Data Filtering: The system retrieves the learner's profile and filter the features required for the collaborative filtering and content filtering. The layer also filters the

courses for which the learner has not provided any rating.

3. Feature Extraction: For content-based approach, we apply feature extraction techniques that unify numerical as well as text features. We apply LDA to extract the features from textual data - course title, course description, and profile description of the learner. These features are represented as vectors. The numerical features are then combined with the textual feature vectors and pass it to the content-based algorithm.

4. Learner Training: In this step we separately train collaborative filtering algorithms (such as Singular Value Decomposition) and content-based algorithms (Deep neural network model).

5. Content Prediction: Finally, we apply the trained model on test set, i.e. new courses posted on the platform. The trained model can be used to predict the rating that the learner will provide to the new courses. The system provides the top-3 recommendations to the learners sorted based on the decreasing order of the rating predicted by the trained model.

5. DATASET

We collected the dataset from Learning & Development team within Accenture through the REST-based services. The dataset consists of learner's profile information and the courses they have enrolled or completed. The dataset consists of 5,000 unique learners and 49,202 unique course content, resulting in total of 2,140,476 enrollments by all learners. The learners have enrolled for multiple courses.

6. EVALUATION AND RESULTS

In this section, we discuss the evaluation of our proposed framework. For collaborative filtering techniques, we considered tuples of <*Learner Id*, *Course Id*, *Rating*> as features. In content-based technique, we considered tuples of <*Skills*, *Country*, *Experience*, *Industry*, *Title*, *Description*, *Rating*, *Category*, *Duration*, *Profile Overview*, *Content Type*>. The features considered for the study are explained in Table 2. These features are passed as an input to the Deep Neural Network model. We used 10-fold cross-validation set up in order to avoid overfitting. Deep neural network models are typically sensitive to the magnitude or scale of features. We apply feature scaling to all the features used as inputs in Deep neural network. The training process will run for a fixed number of iterations through the training dataset called epochs. We used 50 as epoch size. We specified batch size as 10. Batch size is the number of instances that are evaluated in the training set before the weights are updated in the neural network. We applied efficient Gradient Descent algorithm “Adam” [10], an optimizer used to search through different possible weights for the network that minimize loss. Multi-layer Perceptron model requires tuning a number of hyperparameters such as the number of hidden layers, number of neurons in each hidden layer, batch size, epochs, optimizer, activation function etc. We used GridSearch [14] techniques to find the best parameters for a prediction algorithm. It performs exhaustive search over specified parameters for any estimator object. The parameters of the estimator object are optimized by cross validated grid-search. We use “GridSearchCV” library in scikit-learn [12]. We also use trial and experimentation approach to arrive at the optimal number of neurons at each hidden layer that minimizes the overall error. The deep neural network model was implemented using Keras library [11]. In order to assess the effectiveness of the proposed hybrid recommendation model, we considered two evaluation metrics - Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) [13]. Mathematically, they are defined as:

$$MAE(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y_i}| \tag{1}$$

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y_i}|^2} \tag{2}$$

where n is the number of samples, $\hat{y_i}$ is the predicted value of the i^{th} sample and y_i is the corresponding true value.

The performance measures of recommendation techniques are shown in Table 1. For evaluating the proposed framework, we compare the recommendations made by a baseline approach with that of the proposed framework. Learners' skillset is one of the most preferred modes of recommending the courses [6]. We considered a baseline approach that involves recommending courses based on the matching of the learners' skills with the course description. The MAE value of baseline algorithm is 1.853. The prediction made by Deep learning algorithm is better than the Collaborative as well as baseline approach. Deep learning algorithm performs better as it extracts more relevant features from the non-linear function. The recommendations made by the proposed framework performed significantly better as compared to the baseline approach.

7. SYSTEM IMPLEMENTATION

The framework is integrated within our internal web-based learning platform. The system recommends the right set of content to the learners based on their preferences captured either implicitly or explicitly. The platform is integrated with a bot which acts as a virtual buddy for the learner. The bot learns the learner's preference implicitly through their course enrollment or completion history. The bot also captures the learner's preference explicitly by asking the learner. We use DialogFlow [19] to build conversational interface (bot) which provides the natural language understanding services via intent identification. The recommendations are exposed as a REST-based services and integrated via webhook of DialogFlow. The system consists of two major components:

1. **Learning Content Recommendation:** The system recommends the relevant training content to the learners based on learner's history. The system provides the recommendations based on two considerations - learner's personal preference and the preference of other similar learners. The former is referred as “Suggested content based on your interests” and the latter as “Content trending among similar peers”.

Table 1: Performance measures of Hybrid Recommendation techniques

Algorithms	MAE	RMSE
Singular Value Decomposition	0.61	1.00
Slope-One	0.68	1.05
K-Nearest Neighbor	0.613	1.03
Deep Neural Network	0.42	0.66

2. **Similar Learner Community:** The system also helps in finding the community of other similar learners and thus promotes peer learning. The learners can communicate with other similar learners and engage in meaningful discussions.

Due to space limitations we are not able to show screenshot of the system. However, screenshots can be accessed at [23]

8. CONCLUSION AND FUTURE WORK

In this work, we proposed hybrid recommendation framework to build learner's preference. The proposed approach solves the cold start problem often faced by new learners. We applied various predictive modeling techniques to evaluate our recommendation framework. We observed that the proposed framework is able to model the learner's preference quite well. As future work, we will include learner's career path preference for recommending learning content. We also plan to pilot the system to a set of users to evaluate the recommendations.

9. ACKNOWLEDGMENTS

The authors would like to thank Learning & development team for funding the research work and providing huge volumes of data in order to carry out this research work. The authors would also like to thank Tejaswini Pd for helping with the development.

10. REFERENCES

[1] Piao, Guangyuan, and John G. Breslin. "Analyzing MOOC entries of professionals on LinkedIn for user modeling and personalized MOOC recommendations." Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization. ACM, 2016.

[2] Qiu, Jiezhong, et al. "Modeling and predicting learning behavior in MOOCs." Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. ACM, 2016.

[3] Apaza, Rel Guzman, et al. "Online Courses Recommendation based on LDA." SIMBig. 2014.

[4] Lemire, Daniel, and Anna Maclachlan. "Slope one predictors for online rating-based collaborative filtering." Proceedings of the 2005 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2005.

[5] Baek, Jiye, and Jesse Shore. "Promoting student engagement in MOOCs." Proceedings of the Third (2016) ACM Conference on Learning@ Scale. ACM, 2016.

[6] Aher, Sunita B., and L. M. R. J. Lobo. "Combination of machine learning algorithms for recommendation of courses in E-Learning System based on historical data." Knowledge-Based Systems 51 (2013): 1-14.

[7] Bousbahi, Fatiha, and Henda Chorfi. "MOOC-Rec: a case based recommender system for MOOCs." Procedia-Social and Behavioral Sciences 195 (2015): 1813-1822.

[8] Srivastava, Nitish, et al. "Dropout: A simple way to prevent neural networks from overfitting." The Journal of Machine Learning Research 15.1 (2014): 1929-1958.

[9] Nasrabadi, Nasser M. "Pattern recognition and machine learning." Journal of electronic imaging 16.4 (2007): 049901.

[10] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

[11] Chollet, FranÃğois. "Keras: Deep learning library for theano and tensorflow.(2015)."

[12] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." Journal of machine learning research 12.Oct (2011): 2825-2830.

[13] Willmott, Cort J., and Kenji Matsuura. "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance." Climate research 30.1 (2005): 79-82.

[14] Bergstra, James S., et al. "Algorithms for hyper-parameter optimization." Advances in neural information processing systems. 2011.

[15] Hecht-Nielsen, Robert. "Theory of the backpropagation neural network." Neural networks for perception. 1992. 65-93.

[16] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." Journal of machine Learning research 3.Jan (2003): 993-1022.

[17] Wallach, Hanna M. "Topic modeling: beyond bag-of-words." Proceedings of the 23rd international conference on Machine learning. ACM, 2006.

[18] Staubitz, Thomas, and Christoph Meinel. "Collaboration and Teamwork on a MOOC Platform: A Toolset." Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale. ACM, 2017.

[19] https://dialogflow.com/. [Online; accessed 04-Mar-2018].

[20] Onah, Daniel FO, Jane Sinclair, and Russell Boyatt. "Dropout rates of massive open online courses: behavioural patterns." EDULEARN14 proceedings (2014): 5825-5834.

[21] Zepke, Nick, and Linda Leach. "Improving student engagement: Ten proposals for action." Active learning in higher education 11.3 (2010): 167-177.

[22] Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." Computer 42.8 (2009).

[23] https://github.com/kumarabhinav04/LeCoRe.

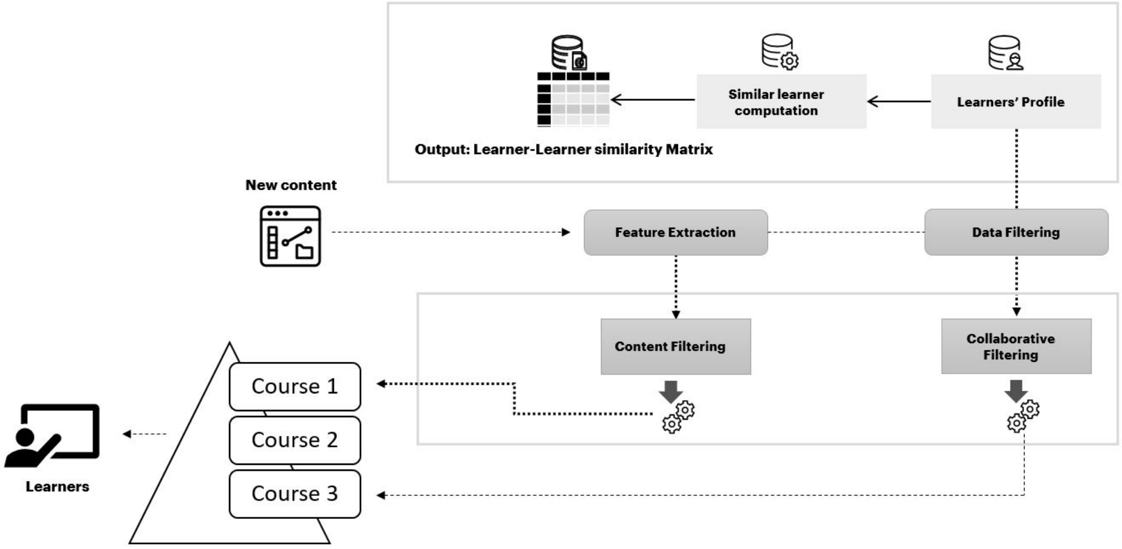


Figure 1: LeCoRe Framework

Table 2: Features considered for the study

Attributes	Description	Entity
Course Id	Unique Id of the course	Course
Title	Textual summary of the course	Course
Description	Textual description of the course	Course
Category	Category of the course e.g. Web development, Mobile development	Course
Duration	Total duration of the course	Course
Content Type	Type of the content e.g. web-based, classroom etc.	Course
Learner Id	Unique Id of the learner	Learner
Country	Country to which learners belong	Learner
Skills	Skills possessed by the learner	Learner
Education	Level of educational degree learner has. We considered five levels of education - High School, Diploma, Bachelor, Masters, and PhD	Learner
Experience	Total years of work experience learners possess	Learner
Profile Overview	Profile description of the learner	Learner
Industry	Industry group (Accenture Vertical) to which the learner belongs to e.g. Financial Services, Health & Public Service etc.	Learner
Project description	Textual description of the projects completed by the learner	Learner
Rating	Score provided by the learner to a course on a scale of 1-5	Learner-Course

Predictive Student Modeling for Interventions in Online Classes

Michael Eagle
TutorGen, Inc.
1037 S. Fort Thomas Ave
Fort Thomas, KY 41075
MichaelJohnEagle@gmail.com

Ted Carmichael
TutorGen, Inc.
1037 S. Fort Thomas Ave
Fort Thomas, KY 41075
ted@tutorgen.com

Jessica Stokes
Western Governors University
4001 South 700 East, ste. 700
Salt Lake City, UT 84107
jessica.stokes@wgu.edu

Mary Jean Blink
TutorGen, Inc.
1037 S. Fort Thomas Ave
Fort Thomas, KY 41075
mjblink@tutorgen.com

John Stamper
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
jstamper@cs.cmu.edu

Jason Levin
Western Governors University
4001 South 700 East, ste. 700
Salt Lake City, UT 84107
jason.levin@wgu.edu

ABSTRACT

As large-scale online classes become more prevalent there is great interest in finding ways to model students at scale in these classes in order to predict outcomes. Student models, if successful, would help determine strong predictors of student success, which would highlight potential causal factors for such success, allowing schools to focus on refinements and interventions that positively impact their student outcomes. In this research, TutorGen has partnered with Western Governors University (WGU), a large online university, and gathered data at scale in order to build exploratory models to predict student outcomes. This paper presents our results so far in successfully identifying students who will pass (or even take) the final exam. We have examined the order in which students take courses, as well as the timing of starting and completing work; our initial analysis reveals that these are strong predictors of course outcomes.

Keywords

Predictive modeling, online education, student interventions, cognitive models, student models, online courseware, feature selection, data visualization, mixed effects modeling, logistic regression.

1. INTRODUCTION

In collaboration with Western Governors University, a large online university, we have been examining large data sets of students' online interactions, which will help provide insight into the way students succeed in course completion. Our initial work has focused on building a set of predictive models looking at success within course and also between courses. Based on student in course data as well as post course assessment, we know learning is occurring within courses. We do know that some students do not pass the final assessment, and these are the students we were most interested in modeling. From our exploratory data analysis and initial models, three important and distinct factors emerged. First, there was a "basic dropout," students who did not pass simply because they stopped being active in the course – not completing assignments or taking tests. Second, was a group termed "late and out of time" that started very late in the semester and appeared to run out of time. And finally, there was group we termed "exam avoidance" that appeared to have mastered enough to pass the final assessment, but for unknown reasons did not attempt the test. We examined some of the properties of each of these types of students and suggest strategies that we plan to implement in order to intervene with each type appropriately.

2. BACKGROUND

Western Governors University (WGU) was founded in 1997 by nineteen governors as a non-profit, competency-based, 100% online university, which has graduated more than 100,000 students. WGU currently has more than 94,000 students across all 50 states. Offering 60 degrees in four colleges supporting high-demand fields such as business, K-12 teacher education, information technology, health professions, WGU's success is founded on their unique learning ecosystem that is student-centric, competency-based, and 100% technology enabled. This approach lowers tuition and provides faster time to graduation.

The WGU competency-based model enables students to leverage their existing knowledge and skills while seeking to improve their opportunities to advance their careers. Students earn college credit by demonstrating what they know and can do rather than basing the credit on seat-time in a course. The curriculum and assessments are defined by career-relevant competencies to accelerate learning according to a student's level of experience. WGU provides the curriculum, formative assessments, and summative evaluation. In addition, the WGU student-centric support services promote success in student learning through disaggregated faculty roles including: program mentors, course instructors, evaluators, and curriculum and assessment developers. This approach allows students to remain at the center of all activities such that the focus is on the learning of each student. The success of the online model that combines competency-based learning models with strong student support is demonstrated by their high student satisfaction ratings, contributing to a higher than average retention rate.

WGU is committed to continually evaluating all aspects of their educational experience in order to enhance their models, content, delivery methods, and student support through proven academic research. This approach helps them make strategic and impactful improvements to student learning outcomes while enhancing the overall student experience (reduced time to achievement, career advancement, learning support, completion/achievement etc.) This drive to measure and identify areas for improvement resulted in a deep data dive evaluating courses that have relatively lower completion rates than others within the same program.

2.1 WGU Course and Term Structure

WGU students begin their degree programs on the first of any month, which begins their first term. A term at WGU is six months in length. Tuition is billed at a flat-rate every term, so students pay for the time, not by credit hour or by course. Students are encouraged to complete as many courses as they can in those six months, resulting in cost savings for students. Students complete courses by passing assessments, demonstrating competency.

2.2 WGU Faculty

The faculty at WGU underpin WGU's unique, student-centric, competency-based approach that places the greatest emphasis on student learning. Learning at WGU is competency-based, the institution does not use typical online classes that are dependent upon fixed schedules or group pacing. Instead, each student is guided and assisted through a personalized learning experience by two primary roles: program mentors and course instructors.

2.2.1 Program Mentors

For each student, the primary faculty support is a personally assigned Program Mentor. The role of the Program Mentor is to provide program instruction, coaching, and support from the moment an individual becomes a student to the time he or she graduates. More specifically, Program Mentors:

- Provide instruction and guidance at the program level.
- Provide information on programs, policies, and procedures.
- Assess students' strengths and development areas to help them develop a plan of study.
- Provide feedback on assessments and recommend learning resources.
- Help students to sustain motivation and maintain on-time progress to their degree.
- Recommend appropriate student services.

This support involves regularly scheduled academic progress conversations weekly and active involvement in other aspects of the student's academic career. While not an expert in all subjects, the Program Mentor guides the student through the overall program and offers coaching, direction, and practical advice.

While there is a default order to degree paths, mentors and students are empowered to personalize the course order. During enrollment each term, the student and program mentor agree upon a set of courses to meet the credit requirement for that term. They set an order, taking any prerequisites into account. Estimated start/end dates are populated for each course assuming the student works on 1 course at a time. (The student may, however, opt to work on multiple courses consecutively.) The program mentor helps guide students' academic activities.

2.2.2 Course Instructors

WGU's Course Instructors are subject matter experts who instruct and support students as they engage specific sections of the WGU curriculum. Their experience and advanced training is specific to the courses they support. They are knowledgeable and can address any issue that might arise related to a course, a learning resource, or an assessment. Specifically, Course Instructors:

- Bring WGU courses of study to life with students via one-to-many or one-to-one forums.
- Provide instructional help (proactively and reactively) and facilitate learning communities.
- Provide content expertise for students who are struggling with course material.

The type and intensity of instructional support varies based on the needs of each student in a particular course, from help with specific questions that arise to more fully engaged tutorial support.

2.3 Assessments at WGU

WGU has developed assessments for each course based on the competencies identified for each course subject. Assessments can take several forms at WGU but follow two main categories: performance assessments and objective assessments.

Performance assessments are embedded throughout the course, such as tests, quizzes, and other assignments, as a way to track progress as students complete the course material. Performance assessments receive qualitative feedback from an assessment team using a standard rubric. Objective assessments are timed and proctored summative exams. Question types may include short answer responses, fill-in-the-blank questions, or multiple choice. With a high-speed Internet connection, a block of uninterrupted time, and a dedicated room with no distractions, students can take these exams at home. During the exam, students are monitored by a live proctor through a webcam provided by WGU. The course evaluated here was of the objective assessment variety.

For objective assessment courses, pre-assessments (also called pre-tests or practice tests) help students and faculty gauge student readiness to take an objective assessment. Pre-assessments measure the same content as the objective assessment, with the same question types, and the same time limit. However, the questions that appear on the pre-assessment will be different from those that appear on the objective assessment.

Both pre-assessment and objective assessment results are provided using four categories: unsatisfactory, approaching competence, competent, and exemplary. A score of "competent" or "exemplary" is required to pass a pre-assessment and/or objective assessment. Exactly what constitutes competence for a given assessment is carefully determined by WGU's Assessment department in concert with a group of experts in the subject matter being assessed.

A student's first attempt on their objective assessment is approved by the program mentor. Mentors can require the completion and pass of a pre-assessment before approval to schedule the objective assessment. Second and subsequent attempts are approved by a course instructor. Course instructors will require the student to complete certain tasks to gauge success on the next attempt before an approval is granted. Students are permitted four attempts for each objective assessment requirement. Any attempt thereafter will need to be approved through the program mentor and course instructor senior leadership.

A WGU course is considered complete when the assessment is passed. For courses with objective assessments, a student with extensive prior knowledge can forgo any interaction with the course materials and move directly to the assessment (typically passing the pre-assessment first). This is true for first and subsequent attempts—so if a student is very close to passing during their first attempt, their second attempt might require very little time and/or effort.

3. RELATED LITERATURE

As large-scale online classes become more prevalent there is great interest in finding ways to model students at scale in these classes in order to predict outcomes. This has been a major area of work in the fields of Educational Data Mining (EDM) and Learning Analytics (LAK) [2] and is leading to methods of executing large scale data experiments in near real-time [9]. Student models, if successful, would help determine strong predictors of student success, which would highlight potential causal factors for such success, allowing schools to focus on refinements and interventions that positively impact their student outcomes. For example, early research at Purdue University presented "academic analytics" tools for predicting at-risk students [4,1]. In [12] the authors develop a "survival model" of student dropouts in a MOOC, determining several significant predictors of dropout behavior. MOOCs, however, present a special case of online courseware, and tend to show quantitatively different outcomes than online courseware with a fixed enrollment, monetary costs to students, and offering course credit and accreditation. In this vein some researchers have looked at similar online environments. In [7] the authors not only developed models to predict student outcomes, but also showed how metrics and visual data provided to instructors can help improve outreach and positive interventions.

In this research, TutorGen has partnered with Western Governors University (WGU), a large, fully accredited online university that offers course credit and an online degree program, and gathered data at scale in order to build exploratory models to predict student outcomes. This paper presents our results so far in successfully identifying students who will pass (or even take) the final exam. We have further examined the order of the courses that students complete and the timing of the course work completed to subsequently show that this order of completion and timing of the work within the semester is a strong predictor of student success.

4. DATA AND METHODS

Our research focused on a single course in the Business school dealing with Finance. The dataset spanning 2016 contained data from over 1,000 students and had low level interaction data of nearly 1 million transactions that was imported into DataShop [8]. In addition to the transaction data, we had practice test data, final summative evaluation data (in the form of Pass, Not Pass, Other), student interactions with the LMS (both with the finance course and other courses), and student summative data from previous courses attempted.

5. ANALYSIS AND DISCUSSION

In order to derive insight about student behavior in the online courses, we used several approaches including: visualization, predictive modeling, and knowledge tracing.

Questions:

1. Are students learning within the course?
2. How are students interacting with the course materials?
3. What are the key differences between the passing and non-passing students?
4. What behaviors describe non-passing students?
5. Can these be used to build an intervention?

To explore question 1, we performed a learning curve analysis on the data in DataShop based on previously defined methods [10]. From these methods, we can visually inspect the learning curves created from low level interaction data tagged at a Knowledge Component (KC) level. From these visualizations we would expect a declining learning curve to emerge as seen in Figure 1. Looking at combined learning curves of all 97 tag KCs in our dataset, we visually saw a declining curve suggesting learning is occurring within the course. Drilling down to individual skills the majority were also classified as "good" in the DataShop learning curve interface suggesting learning is occurring.

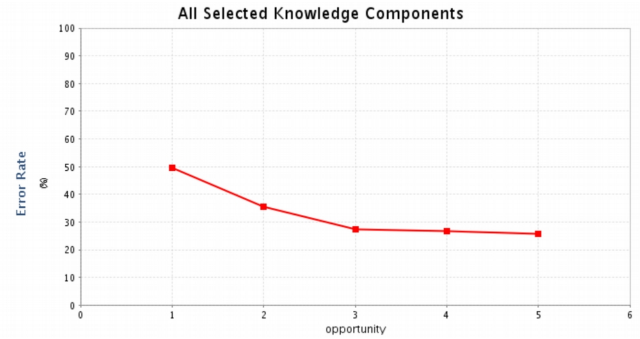


Figure 1. A sample learning curve of all KCs. The y-axis represents error rate and the x-axis represents all student opportunities to apply each skill. If learning is occurring we expect to see a declining curve as seen.

To examine question 2, we did an exploration of the activities that students did within their course. We did note that the particular course we were exploring was one of the most difficult and had lower than average completion rates. We used the finest-grained level of data available. This data included the student's step by step actions in the online system. We constructed a visualization to represent student behavior across time, as well as how they performed on practice tests and final assessments.

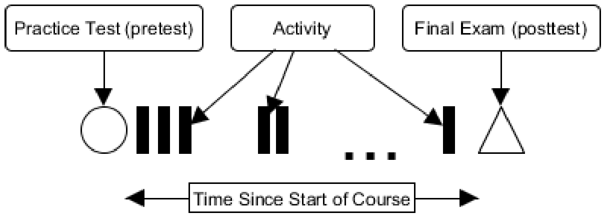


Figure 2. Timeline for an individual student. Vertical bars represent a course activity session, and circles and triangles represent the practice test and final exam attempts.

Figure 2 shows a timeline for an individual student, vertical bars represent a course activity session (reading, practice problems, etc. performed within the same time online session) Circles and Triangles represent the practice test and final exam attempts. This provides a high-level view of student work across time, as well as visually representing the student testing behaviors. We used color (blue and orange) to encode passing and non-passing students, figure 4 shows typical behavior of passing students while figure 5 shows typical behavior of some non-passing students.

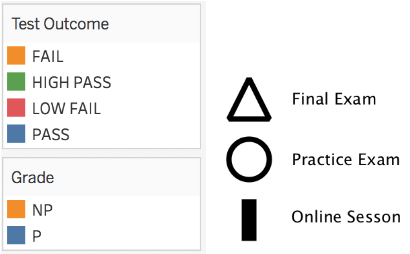


Figure 3. Key for the work timeline visualization.

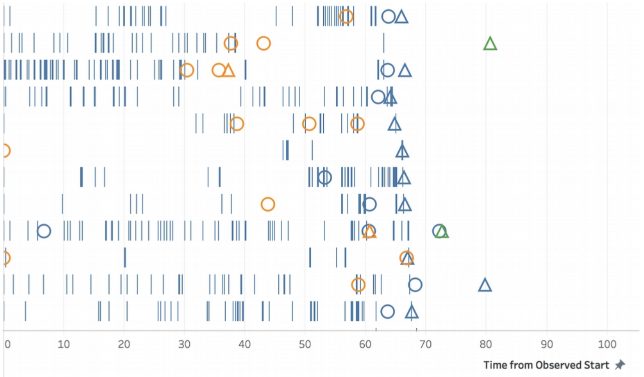


Figure 4. Typically, students engage with the course for a while before taking the practice test, depending on the results of that test they take and pass the final exam. It is rare for students to take the practice test before starting some of the coursework.

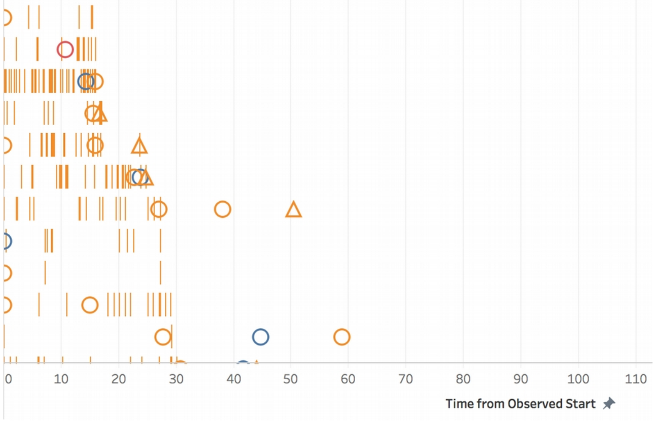


Figure 5. A common behavior for non-passing students is to dropout fairly soon after starting the course. Students have multiple opportunities to take the final exam, however many of the non-passing students do not take the final multiple times.

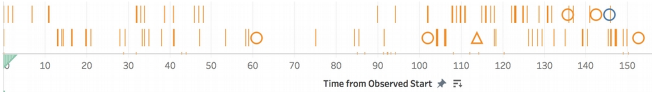


Figure 6. Long fail behavior, note the late pass on the practice exam with no attempt on the final exam. Not all failing students dropout quickly after starting, some students do work throughout a term. Interestingly, some students pass the practice exam but sim

These exploratory visualizations helped to highlight three important predictive features of this data: deviation from planned course start, action density once started, and the course order. For example, some students would quickly work through the material while others would spread it out.

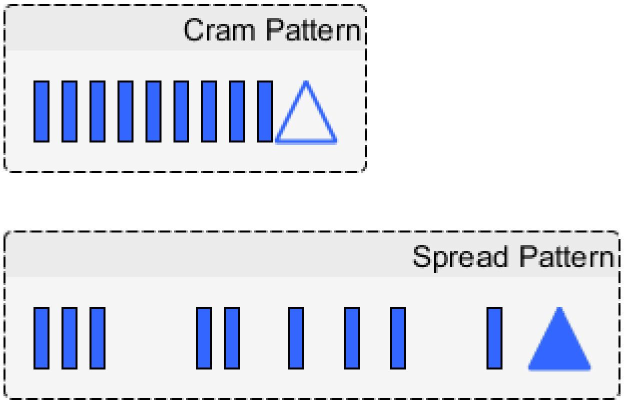


Figure 7. Cram pattern of activity vs. a spread pattern of activity. How students planned and executed their coursework proved to be predictive of success.

Students can choose when to schedule courses throughout a 180-day long term. We found that both planned start time, as well as deviation from that start time to be an important factor between passing and non-passing students. This information would help us to build a predictive model, and allow for more targeted interventions, as discussed in the next section.

5.1 Predictive Model

In order to explore potential interventions, we aimed to create predictive models for test performance and course grade. We developed a predictive model for student test performance and overall course performance. In order to be useful in a live environment, we needed an updating model that makes a prediction based on the continuously updating activity data. We also needed to address the fact that students are able to choose when to take the assessment tasks.

For each session, we want to make a prediction about the student performance on the next test. We refer to the sessions between (or in the case of the first test taken, all sessions before) testing opportunities as assessment windows. Our model will use the previously observed data to predict the performance on the next testing opportunity.

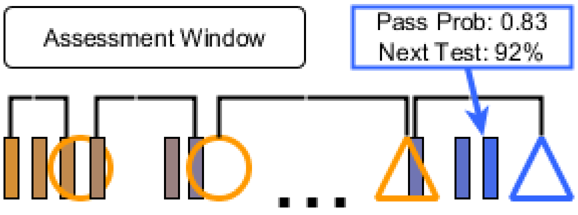


Figure 8. Next test performance is predicted using information from assessment windows proceeding testing opportunities.

We integrate the results of these models into our visualization, which takes the form of an orange to blue gradient on colorization with extra information available in tool tips.

We predict for each session the probability that the student will ultimately pass the course, as well as their predicted score on the next testing opportunity. We use a logistic regression for the pass prediction and a mixed effects model for the next test prediction. The overall performance of the pass prediction model was fair with 69% accuracy; when using an 80% cutoff value on the pass prediction we had a positive prediction value (Recall) of about .7 and a negative prediction value (Precision) of .66. The most important features include observed test performance, total number of observed actions, total time engaged in the materials, the accuracy the lesson activities, the total number of sessions, and the amount of time from the planned start of the course.

The next test prediction model also had fair performance with an overall RMSE of 6.92, which improves slightly as more data is added to a value of 6.6 until around the 20th session, after which the RMSE gradually moves to 6.95. The primary predictive features were previous test performance, total observed transactions, accuracy on lesson activities, time between sessions, and the amount of time between the planned start of the course.

Exploration of the model predictions and the visualization revealed that a number of students who are expected to pass the next assessment, simply never attempt the final. For students that ultimately do not pass the course, roughly 32% never attempt the final assessment (19% of failing students take neither the pretest or final.) Our model indicates that 35% of failing students would have been likely to receive a passing grade on their next test (30% if we only include students with at least one final exam attempt.) This is evidence of potential test anxiety [5] or avoidance of demonstrating a lack of ability [6]. Avoiding tests is not an uncommon occurrence in low-stakes tests [11]. See Table 1 for the complete breakdown of unsuccessful course explanations.

Table 1. Explanations or potential explanations, for students who did not pass the examined courses

Reason	Proportion
Quit Early, low use of resources	13%
Has Activity, Complete Testing Avoidance	4%
Has Activity, Predicted Pass w/o Test	35%
Ran out of time in term (Started very late)	8%
Low Activity	22%
Not Explained	18%

5.2 Intervention Opportunities

There are several opportunities for developing interventions that can improve student outcomes. There are three primary targets: Instructors, Mentors, and Students. Rather than target the students directly, we will focus on providing information to the Program Mentors. By providing visualizations like the ones above, we can allow the mentors to create unique advice to the students. For example, a mentor can provide encouragement to take an attempt on the final due to the next test prediction metric. Targeting the students who seem to be avoiding the final assessment is an area that could provide great impact to student outcomes. The challenge here is to balance flexibility for students - a key attribute of competency-based education [3] - with enough structure and support, in order to optimize student performance. For example, while a flexible timeframe for completing coursework is a hallmark of competency-based learning, it may prove that many students need some structure and prompting in order to compel completion rates.

6. CONCLUSIONS AND FUTURE WORK

WGU has a unique structure for online educational programs. Exploration of student data revealed that students generally make use of the course resources and learning materials, but will sometimes fail courses due to course scheduling and failing to adhere to their planned start dates. More importantly, a significant proportion of failing students would likely pass if they would attempt to take the final assessment. We propose interventions targeting the program mentors, rather than students, in order to explore methods of addressing the scheduling, activity, and test avoidance issues that make up the majority of the reasons students fail to pass courses. The end goal is to proactively advise students, course instructors, and student mentors with relevant just-in-time information in order to insert and test appropriate interventions.

7. ACKNOWLEDGMENTS

This research was supported through funding from the National Science Foundation Small Business Innovative Research Award #1534780.

REFERENCES

[1] Baepler, P., and Murdoch, C. Academic Analytics and Data Mining in Higher Education. International Journal for the Scholarship of Teaching and Learning 4(2) (2010).

[2] Baker, R. S. J. D., Duval, E., Stamper, J., Wiley, D., & Buckingham Shum, S. (2012). Panel: educational data mining meets learning analytics. In *Proceedings Of International Conference On Learning Analytics & Knowledge* (Vol. 2).

[3] Bral, C. and Cunningham, J. (2016), Foundations of quality in competency-Based programs: Competencies and assessments. The Journal of Competency-based Education, 1: 118–121. doi: 10.1002/cbe2.1027.

[4] Campbell, J., De Blois, P. B., and Oblinger, D.G. Academic analytics. A New Tool for a New Era. EDUCAUSE Review 42(4): 42–57 (2007).

[5] Elliot, Andrew J., and Holly A. McGregor. "Test anxiety and the hierarchical model of approach and avoidance achievement motivation." *Journal of Personality and social Psychology* 76.4 (1999): 628.

[6] Middleton, Michael J., and Carol Midgley. "Avoiding the demonstration of lack of ability: An underexplored aspect of goal theory." *Journal of educational psychology* 89.4 (1997): 710.

[7] Smith, V. C., Lange, A., & Huston, D. R. (2012). Predictive modeling to forecast student outcomes and drive effective interventions in online community college courses. *Journal of Asynchronous Learning Networks*, *16*(3), 51-61.

[8] Stamper, J., Koedinger, K., d Baker, R. S., Skogsholm, A., Leber, B., Rankin, J., & Demi, S. (2010, June). PSLC DataShop: A data analysis service for the learning science community. In *International Conference on Intelligent*

Tutoring Systems (pp. 455-455). Springer, Berlin, Heidelberg.

[9] Stamper, J. C., Lomas, D., Ching, D., Ritter, S., Koedinger, K. R., & Steinhart, J. (2012). The Rise of the Super Experiment. *International Educational Data Mining Society*.

[10] Stamper, J., Koedinger, K., & Mclaughlin, E. (2013, July). A comparison of model selection metrics in DataShop. In *Educational Data Mining 2013*.

[11] Swerdzewski, Peter J., J. Christine Harmes, and Sara J. Finney. "Skipping the test: Using empirical evidence to inform policy related to students who avoid taking low-stakes assessments in college." *The Journal of General Education* (2009): 167-195.

[12] Yang, D., Sinha, T., Adamson, D., & Rosé, C. P. (2013, December). Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In *Proceedings of the 2013 NIPS Data-driven education workshop* (Vol. 11, pp. 14).

GritNet: Student Performance Prediction with Deep Learning

Byung-Hak Kim, Ethan Vizitei, Varun Ganapathi
Udacity
2465 Latham Street
Mountain View, CA 94040
{hak, ethan, varun}@udacity.com

ABSTRACT

Student performance prediction - where a machine forecasts the future performance of students as they interact with online coursework - is a challenging problem. Reliable early-stage predictions of a student's future performance could be critical to facilitate timely educational interventions during a course. However, very few prior studies have explored this problem from a deep learning perspective. In this paper, we recast the student performance prediction problem as a sequential event prediction problem and propose a new deep learning based algorithm, termed GritNet, which builds upon the bidirectional long short term memory (BLSTM). Our results, from real Udacity students' graduation predictions, show that the GritNet not only consistently outperforms the standard logistic-regression based method, but that improvements are substantially pronounced in the first few weeks when accurate predictions are most challenging.

Keywords

Student performance prediction, Deep learning for education, Educational data mining, Learning analytics

1. INTRODUCTION

Education is no longer a one-time event but a lifelong experience. One reason is that working lives are now so lengthy and fast-changing that people need to keep learning throughout their careers [3]. While the classic model of education is not scaling to meet these changing needs, the wider market is innovating to enable workers to learn in new ways. Massive open online courses (MOOCs), offered by companies such as Udacity and Coursera, are now focusing much more directly on courses that make their students more employable. At Coursera and Udacity, students pay for short programs that bestow *microcredentials* and *Nanodegrees* in technology-focused subjects such as self-driving cars and Android. Moreover, universities are offering online degrees to make it easier for professionals to access opportunities to develop their skills (e.g., Georgia Tech's Computer Science Master's degree).

However, broadening access to cutting-edge vocational subjects does not naturally guarantee student success [2][1]. In a classic classroom, where student numbers are limited, various dimensions of interactions enable the teacher to quite effectively assess an individual student's level of engagement, and anticipate their learning outcomes (e.g., successful completion of a course, course withdrawals, final grades). In the world of MOOCs, the significant increase in student numbers makes it impractical for even experienced *human* instructors to conduct such individual assessments. An automated system, which accurately predicts how students will perform in real-time, could possibly help in this case. It would be a valuable tool for making smart decisions about when to make live educational interventions during the course (and with whom), with the aim of increasing engagement, providing motivation and empowering students to succeed.

The student performance prediction problem has been partly studied within the learning analytics and educational data mining communities in the form of the student dropout (or completion) prediction problem (which is an important subclass problem of the student performance prediction problem). Most previous works can be divided into two approaches:

- The first traditional approach principally relies on generalized linear models, including logistic regression, linear SVMs and survival analysis (see [10] for a thorough summary). Each model considers different types of behavioral and predictive features extracted from various raw activity records (e.g., clickstream, grades, forum, grades).
- The second emerging approach involves an exploration of neural networks (NN). Few prior works explore deep neural network (DNN) model [10], recurrent neural network (RNN) model [6] and convolutional neural networks (CNN) followed by RNN [9]. However, all of these new models, so far, have shown primitive performance. This is mainly because the models still rely on feature engineering to reduce input dimensions which appears to limit one to develop larger (i.e., better) NN models.

Student activity records collected from different courses often have various lengths, formats and content, so that features that are effective in one course might not be so in another. Even carefully designed feature dimensions are usually constrained to be small[2]. Both of these deficiencies produce inputs that are, so far, too restricted to tap the full

[1] HarvardX and MITx have reported that only 5.5% of people who enroll in one of their online courses earn a certificate.

[2] In past works, DNN of width 5 [10] and LSTM of 20 cell dimensions [6] are used.

benefits of sequential deep learning models. To avoid the deficiencies of prior works, GritNet takes students' learning activities across time as raw input (see Section 2.3.1) and (implicitly) searches for parts of an event embedding sequence that are most discrminative to predicting a student's performance without having to engineer those parts as an (explicit) input feature (see Section 2.3.2).

In the remainder of this paper, we introduce the basic GritNet model in Section 2, followed by the Udacity data and training discussions in Section 3. In Section 4, we demonstrate the performance of GritNet via experimental results and give conclusions in Section 5.

2. GritNet

2.1 Problem Formulation

The task of predicting student performance can be expressed as a sequential event prediction problem [8]: given a past event sequence $\mathbf{o} \triangleq (o_1, \ldots, o_T)$ taken by a student, estimate likelihood of future event sequence $\mathbf{y} \triangleq (y_{T+D}, \ldots, y_{T'})$ where $D \in \mathbb{Z}_+$.

In the form of online classes, each event o_t represents a student's action (or activities) associated with a time stamp. In other words, o_t is defined as a paired tuple of (a_t, d_t). Each action a_t represents, for example, "a lecture video viewed", "a quiz answered correctly/incorrectly", or "a project submitted and passed/failed", and d_t states the corresponding (logged) time stamp.

Then, log-likelihood of $p(\mathbf{y}|\mathbf{o})$ can be written as Equation 1, given fixed-dimensional embedding representation υ of $\mathbf{o}$.

$$\log p(\mathbf{y}|\mathbf{o}) \simeq \sum_{i=T+D}^{T'} \log p(y_i|\upsilon) \qquad (1)$$

The goal of each GritNet is, therefore, to compute an individual log-likelihood $\log p(y_i|\upsilon)$, and those estimated scores can be simply added up to estimate long-term student outcomes.

2.2 Baseline Model

In order to assess how much added value is brought by the GritNet, logistic regression is used as a baseline model. Here, we use the bag of words (BoW) model to represent each student's past event sequence $\mathbf{o}$. After transforming all students' activities into a BoW, we count the number of times each unique activity appears in $\mathbf{o}$.

Let fixed-dimensional feature representation υ of $\mathbf{o}$ be an N-dimensional feature vector where $\upsilon_j \in \mathbb{Z}_{\geq 0}$. Given υ, logistic regression models $\log p(y_i|\upsilon)$ as follows:

$$\log p(y_i = 1|\upsilon; \theta) = \frac{1}{1 + \exp\left(-\theta^T \upsilon\right)}, \qquad (2)$$

where $\theta \in \mathbb{R}^N$ are the parameters of the logistic regression model. For M training instances $\left\{\left(\upsilon^{(k)}, y^{(k)}\right)\right\}_{k=1}^{M}$, L_2 regularized logistic regression finds the parameters θ that solve the following optimization problem:

$$\arg\max_{\theta} \sum_{k=1}^{m} \log p(y^{(k)}|\upsilon^{(k)}; \theta) + \alpha\|\theta\|_2. \qquad (3)$$

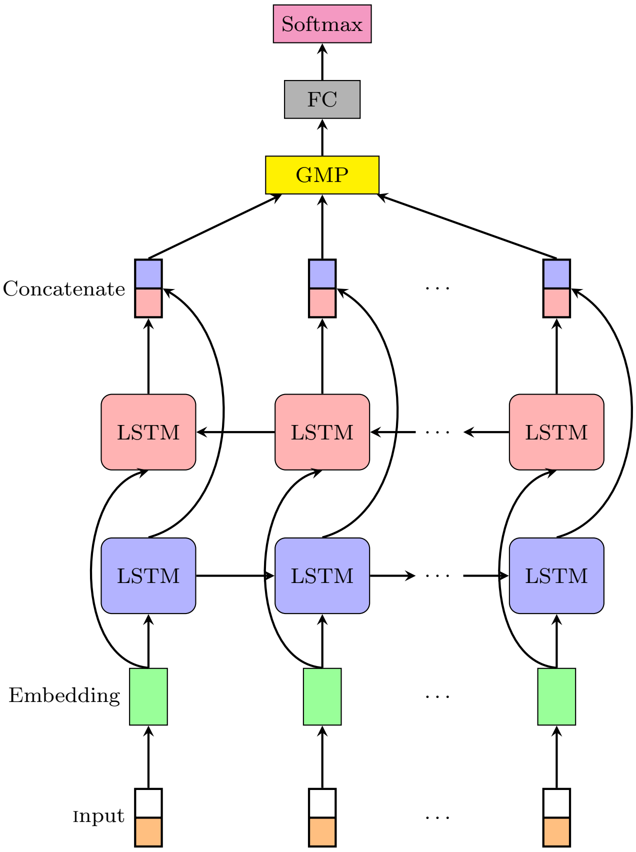


Figure 1: Architecture of a GritNet for the student performance prediction problem as described in Section 2.3.

Often, it will be convenient to consider L_1 regularized logistic regression instead of Equation 2 to handle irrelevant features [7]. We noticed even simpler feature selection methods (e.g., Chi-Square score based), combined with L_2 regularized logistic regression, provides similar results as the L_1 based.

2.3 GritNet Architecture

2.3.1 *Input Representation*

In order to feed students' raw event records into the GritNet, it is necessary to encode the time-stamped logs (ordered sequentially) into a sequence of fixed-length input vectors[3]. We do this simply by *one-hot encoding*. A one-hot vector $\mathbb{1}(a_t) \in \{0, 1\}^L$, where L is the number of unique actions and j-th element defined as:

$$\mathbb{1}(a_t)_j \triangleq \begin{cases} 1 & \text{if } j = a_t \\ 0 & \text{otherwise} \end{cases}, \qquad (4)$$

is used to distinguish each activity a_t from every other. Then, we connect one-hot vectors of the same student into a long vector sequence to represent the student's whole sequential activities in $\mathbf{o}$.

[3] GritNet does not need manual feature selections [6] or time-series input aggregations per normalized time intervals [9].

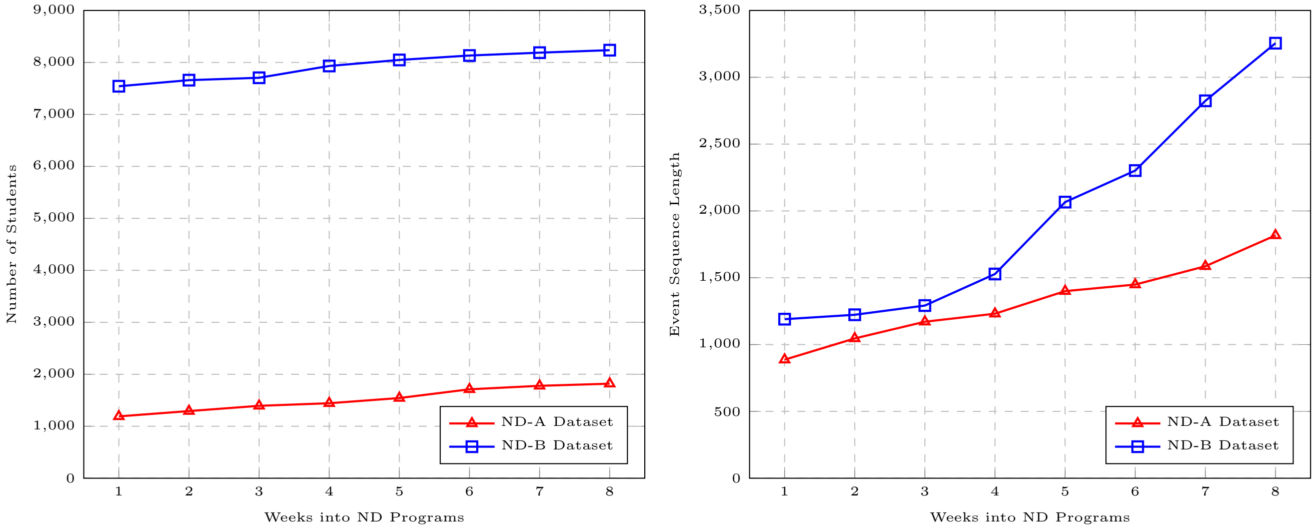


Figure 2: Student and event characteristics of two Udacity Nanodegree program datasets (ND-A and ND-B) by week: size of dataset (left) and maximum event sequence length (right). The reason the number of students climbs over time is that we only include students who have interacted with their mentor, so if they do not interact in the first couple weeks they are not included early on (left). As a student progresses, the accumulated event sequence gets longer (right).

While this encoding method preserves ordering information, in contrast to the BoW method (see Section 2.2), it has a limitation in capturing students' learning speed. Varying learning speed is an important piece of time-dependent information, reflecting a student's progress and/or the course content's difficulty. Since directly employing each time stamp d_t will increase the input space too fast, we define the discretized time difference between adjacent events as[4]:

$$\Delta_t \triangleq d_t - d_{t-1}. \tag{5}$$

Then, one-hot encode Δ_t into $\mathbb{1}(\Delta_t)$ and connect them with the corresponding $\mathbb{1}(a_t)$ to represent $\mathbb{1}(o_t)$ as:

$$\mathbb{1}(o_t) \triangleq [\mathbb{1}(a_t); \mathbb{1}(\Delta_t)]. \tag{6}$$

Lastly, we pre-pad the output sequences shorter than the maximum event sequence length (of a given training set) with all $\mathbf{0}$ vectors.

2.3.2 *Model Architecture*

The core of our GritNet model is the embedding [1], BLSTM [4] and GMP [5] layers trained to ingest past student events and predict a log likelihood of a future one. The first embedding layer[5] learns an embedding matrix $\mathbf{E}^o \in \mathbb{R}^{E\times|O|}$, where E and $|O|$ are the embedding dimension and the number of unique events (i.e., input vector $\mathbb{1}(o_t)$ size), to convert an input vector $\mathbb{1}(o_t)$ onto a low-dimensional embedding v_t defined as:

$$v_t \triangleq \mathbf{E}^o \mathbb{1}(o_t). \tag{7}$$

This event embedding v_t is then passed into the BLSTM and the output vectors are formed by concatenating each forward and backward direction outputs. Next, a GMP layer is added before the output layer. With the GMP layers, GritNet learns to focus the most relevant part of the event embedding sequence while ignoring the rest. This GMP operation seems crucial in boosting prediction power, particularly for imbalanced data provided without any feature engineering[6].

The GMP layer output is, ultimately, fed into a fully-connected layer and a softmax (i.e., sigmoid) layer sequentially to calculate the log-likelihood $\log p(y_i|v)$. The complete GritNet architecture is illustrated in Figure 1.

3. DATA AND TRAINING

3.1 Udacity Data

We benchmarked our methods on the student datasets of two Udacity Nanodegree (ND) programs: ND-A and ND-B. These two ND programs were selected specifically because they diverge from each other along many axes. For example, ND-A curriculum has a lower expectation of prior technical knowledge and a relatively higher graduation rate than ND-B.

[4] For the Udacity data described in Section 3.1, we use day to represent inter-event time intervals.

[5] With an embedding layer which provides a dense representation for an event, GritNet achieves an improved performance on the Udacity dataset. Furthermore, after training, similar events appear to be closer in the embedding event space.

[6] We empirically find that vanilla BLSTM (without the GMP layer) on the (imbalanced) Udacity datasets does not yield comparable results as shown in Figure 3. A GMP layer appears to combat this imbalanced data issue effectively by ensuring training errors back-propagate only to the network weights corresponding to the most discriminative part within the event embedding sequence.

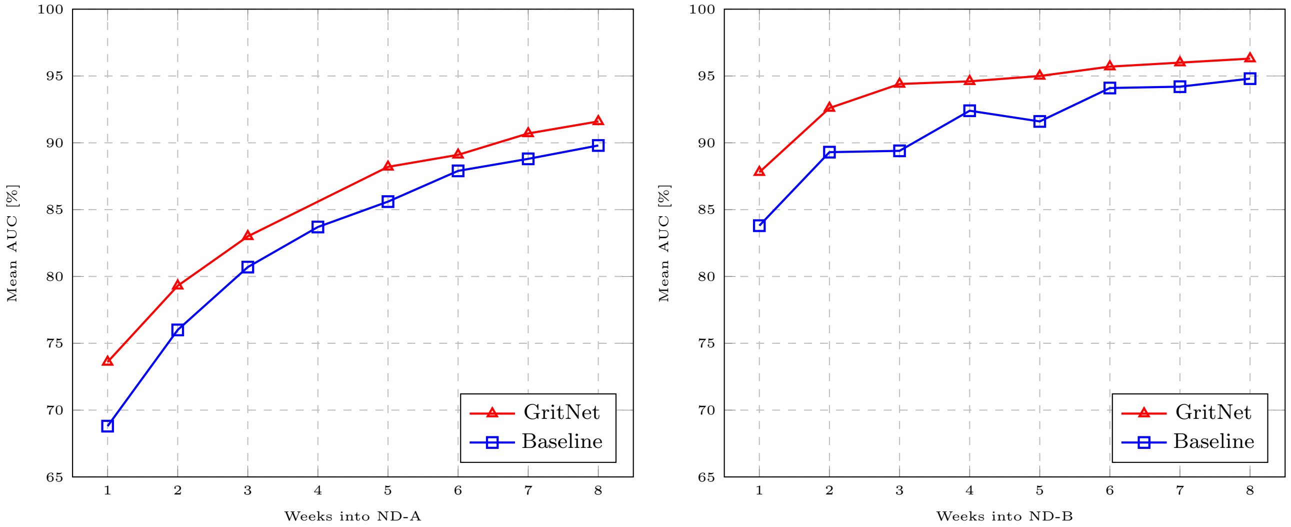


Figure 3: Student graduation prediction accuracy comparisons of the GritNet vs baseline models in terms of mean AUC (over all five folds) on two Udacity Nanodegree program datasets: ND-A (left) and ND-B (right). GritNet provides 5.3% abs (7.7% rel) accuracy improvements at week 1 for ND-A dataset (left). Notice that for ND-B dataset, the baseline model requires eight weeks of student data to achieve the same performance as the GritNet is able to achieve with only three weeks of student data (right).

In both programs, graduation is defined as completing each of the required projects in a ND program curriculum with a passing grade. When a user officially graduates, their enrollment record is annotated with a time stamp, so it was possible to use the presence of this time stamp as the target label. Users have to graduate before 2017-09-30 to be considered as successfully graduated.

Each ND program's curriculum contains a mixture of video content, written content, quizzes, and projects. Note that it is not required to interact with every piece of content or complete every quiz to graduate. See below for detailed characteristics of each dataset used for this study.

- **ND-A Dataset**: From the students who enrolled in ND-A program (from 2017-03-07 to 2017-09-30), we selected 1,853 students who had actively engaged with their classroom mentor (believing these to be the students exhibiting full engagement with the curriculum overall). This set of 1,853 students includes 777 students who graduated, yielding a graduation rate of 41.9%. The length of each student's events streams ranges from 0 to 4,175 events, with an average of 536 events. The curriculum for ND-A program contains 9 projects, 1,025 unique content pages to visit, and 77 quizzes to attempt.
- **ND-B Dataset**: As prescribed above, we selected 8,301 students who actively engaged with their classroom mentor from the students who enrolled in ND-B program (from 2016-06-20 to 2017-09-30). This set of 8,301 students includes 1,005 students who graduated, yielding a graduation rate of 12.1%. The length of each student's event streams ranges from 1 event to 4,554 events, with an average of 242 events. The curriculum for ND-B program is composed of 6 projects, 668 unique content pages, and 347 quizzes.

For both datasets, an event represents a user taking a specific action (e.g., watching a video, reading a text page, attempting a quiz, or receiving a grade on a project) at a certain time stamp. Some irrelevant data is filtered out during preprocessing, for example, events that occur *before* a user's official enrollment as a result of a free-trial period. It should be noted that no personally identifiable information is included in this data and student equality is determined via opaque unique ids.

3.2 Training

We learned that the GritNet models are fairly easy to train. The training objective is the negative log likelihood of the observed event sequence of student activities under the model. The binary cross entropy loss is minimized[7] using stochastic gradient descent on mini-batches.

In our experiment, the BLSTM with forward and backwards LSTM layers containing 128 cell dimensions per direction is used. Embedding layer dimension was grid-searched for the best parameters based on the dataset: from 1024 to 3584 for ND-A set and from 1024 to 5120 for ND-B set. A dropout rate, ranged from 10 to 20%, applied to the BLSTM output and worked well for both datasets to prevent overfitting during training with a mini-batch size of 32.

For both baseline and GritNet models, we trained a different model for different weeks, based on students' week-by-week event records, to predict whether each student was likely to

[7] In this case, minimizing the binary cross entropy is equivalent to maximizing the log likelihood.

graduate. Figure 2 shows the number of students and the (longest) event sequence length of a student, both observed at each week.

4. PREDICTION PERFORMANCE

4.1 Evaluation Measure

To demonstrate the benefits of the GritNet, we focused on student graduation prediction. Since the true binary target label (1: graduate, 0: not graduate) is imbalanced (i.e., number of 0s outweighs number of 1s), accuracy is not an appropriate metric. Instead, we used the Receiver Operating Characteristic (ROC) for evaluating the quality of the GritNet's predictions. An ROC curve was created by plotting the true positive rate (TPR) against the false positive rate (FPR). In this task, the TPR is the percentage of students who graduate, which the GritNet labels positive, and the FPR is the percent of students who do not graduate, which the GritNet incorrectly labels positive.

The accuracy of each system's prediction was measured by the area under the ROC curve (AUC) which scores between 0 and 100% (the higher, the better) − with random guess yielding 50% all the time. We used 5-fold student level cross-validation, while ensuring each fold contained roughly the same proportions of the two groups (graduate and non graduate) of students.

4.2 Results

For fair comparisons, the baseline performance was optimized by sweeping α values (in Equation 3) at each week[8]. The GritNet also required slight hyper-parameter optimization (e.g., embedding dimension as prescribed in Section 3.2) for the optimal accuracy at each week.

We have shown that the GritNet really does improve the student graduation prediction accuracy across weeks. From the prediction results on both Udacity datasets in Figure 3, we clearly see that the performance is similar between the baseline and GritNet models after receiving eight weeks of data about a given student. However, the GritNet is able to achieve significant prediction-quality improvements within the first few weeks of the student experience.

Specifically, the GritNet was able to attain superior performance by more than 5.0% abs on both ND-A dataset (at week 1) and ND-B dataset (at week 3). Moreover, on ND-B dataset, the baseline model required a wait of two months to reach the prediction accuracy that the GritNet showed within three weeks. We believe this is a crucial advantage of the GritNet, creating a quickly adaptable but accurate metric to estimate long-term student outcomes to accelerate the student feedback loop (which typically takes a few months from enrollment to iterate).

5. CONCLUSION

In this paper, we have successfully applied deep learning to the challenging student performance prediction problem which, so far, has not been fully exploited. In contrast to prior work, we formulated the problem as a sequential event prediction problem, introduced a new algorithm called the GritNet to tackle the problem, and demonstrated the superiority of the GritNet using student data from Udacity's Nanodegree programs.

Two novel properties of the GritNet are that (1) it does not need any feature engineering (it can learn from raw input) and (2) it can operate on any student event data associated with a time stamp (even when highly imbalanced). For future work, we anticipate that incorporating indirect data (e.g., student board activity, interactions with mentors) into the GritNet will potentially further improve the GritNet's impressive performance.

6. REFERENCES

[1] Y. Bengio, R. Ducharme, and P. Vincent. A neural probabilistic language model. In *Advances in Neural Information Processing Systems 13 (NIPS 2000)*, pages 932–938, 2001.

[2] I. Chuang and A. Ho. HarvardX and MITx: Four years of open online courses, Fall 2012-Summer 2016. *SSRN Electronic Journal*, 2016.

[3] Economist. Equipping people to stay ahead of technological change - Lifelong learning. *A Special Report On Lifelong Learning: How to Survive in the Age of Automation*, 2017.

[4] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional LSTM networks. In *2005 International Joint Conference on Neural Networks (ICJNN'05)*, pages 23–43, 2005.

[5] J. D. Keeler, D. E. Rumelhart, and W. K. Leow. Integrated segmentation and recognition of hand-printed numerals. In *Advances in Neural Information Processing Systems 3 (NIPS 1990)*, pages 557–563, 1991.

[6] F. Mi and D.-Y. Yeung. Temporal models for predicting student dropout in massive open online courses. In *Proceedings of 15th IEEE International Conference on Data Mining Workshop (ICDMW 2015)*, pages 256–263, Atlantic City, New Jersey, 2015.

[7] A. Y. Ng. Feature selection, L1 vs. L2 regularization, and rotational invariance. In *Proceedings of the 21st International Conference on Machine Learning (ICML 2004)*, pages 78–85, Banff, Alberta, Canada, 2004.

[8] C. Rudin, B. Letham, A. Salleb-Aouissi, E. Kogan, and D. Madigan. Sequential event prediction with association rules. In *24th Annual Conference on Learning Theory (COLT 2011)*, pages 615–634, 2011.

[9] W. Wang, H. Yu, and C. Miao. Deep model for dropout prediction in moocs. In *Proceedings of the 2nd International Conference on Crowd Science and Engineering (ICCSE 2017)*, pages 26–32, Beijing, China, 2017.

[10] J. Whitehill, K. Mohan, D. Seaton, Y. Rosen, and D. Tingley. Delving deeper into mooc student dropout prediction. *arXiv preprint arXiv:1702.06404*, 2017.

[8] The optimized results are quite strong such that initially explored NN models (e.g., DNN, CNN-BLSTM) on the same BoW input features did not yield big win over the baseline.

Limitations of Natural Language Processing Tools for Data Mining: Text Length and Grade Level Differences

Scott A. Crossley
Georgia State University
Atlanta, GA 30303
scrossley@gsu.edu

ABSTRACT

NLP tools have demonstrated themselves to be an important component of educational data mining studies with an increasing number of studies published yearly. However, there is no agreement on the effect that text size has on NLP analyses by feature, text type, domain, and/or level of writer. This study provides evidence that NLP tools provide differential associations between linguistic features and student outcomes based on text size and level within an online learning platform focused on math instruction.

Keywords

Natural language processing (NLP), text length, math success, on-line learning, syntactic complexity, lexical sophistication

1. INTRODUCTION

Natural language processing (NLP) tools are becoming more common in educational data mining (EDM) research. This research includes investigations into learning behaviors in intelligent tutoring systems (ITSs) that teach writing [1] and reading strategies [2], massive open online courses [3], and online tutoring systems [4, 5]. NLP tools, like many other analytic tools, can be extremely powerful and provide exciting new insights into learning and instruction. However, they can also be misused based on poor understandings of the fundamental nature of language, the manner in which the tools calculate linguistic features, and how language is acquired and used by humans. A chief concern in educational data mining is that many language samples produced by learners within instructional systems are short and may not provide enough linguistic coverage to accurately examine the distributional properties of students' language skills. If NLP tools are used to generate a linguistic profile of a student using only a small sample of a student's language, that profile may contain more statistical randomness than actual linguistic information. In such cases, the profile developed will not accurately reflect the student and may rather reflect the task, the student's linguistic performance within a small window of time, or neither.

For instance, educational researchers may be interested in assessing students' reading comprehension as reflected in self-explanations [6, 7] and question generation [8]. However, both tasks elicit only a sentence or two of student self-generated text as do generated questions. While the models developed from these short texts may be predictive and generalizable to a task, it is not clear that they provide a true profile of learners' language knowledge (i.e., learners' language proficiency or ability). What is clear is that learning models for self-explanations derived from language data seem to benefit from longer text samples [6, 7]. These studies along with a number of other studies examining text length outside of educational settings [9, 10, 11] seem to support the notion that longer texts provide stronger and more reliable NLP results. However, a study examining how student language samples of different text lengths moderate the strength of associations between NLP indices and learning variables is missing. In addition, no studies to our knowledge, have examined the potential for NLP indices to differ based on the age or cognitive maturity of learners. Such differences are likely in light of evidence that shows differential language acquisition stages based on age. For instance, research generally supports the notion that children move from producing single words (i.e., a holophrastic stage) around the age of one-year, to two words that are semantically and syntactically related around the age of two-years, and then longer utterances that shared syntactic structures with adult production after three years of age [12].

Thus, the purpose of this study is to examine differences in association strength between NLP indices and a learning variable based on differential text length and grade level. We focus specifically on language data taken from student e-mails in an on-line math tutoring system. We examine how the strength of associations change between a number of attested NLP indices and math scores within the system as a function of text length (texts greater than 50 words, texts greater than 100 words, and texts greater than 150 words) and as a function of grade level (2nd and 3rd grades, 4th and 5th grades, and 2nd-5th grades). Our goal is to examine if text length and grade level influence the strength of association between the NLP indices and math scores.

2. METHOD

2.1 Online Learning System

The data used in this study came from Reasoning Mind's *Foundations* product. RM *Foundations* is a blended learning mathematics program that is used primarily in grade levels 2-5. *Foundations* allows students to learn math concepts at their own pace within an engaging, animated world. Of interest in this study are problems that address basic math knowledge and skills for each objective, which comprise the lowest level of knowledge within the system. These problems address basic knowledge and skills related to the objective covered within the *Foundations* system. The problems are relatively simple and typically require a single step to solve. We used these problems as our benchmark for math success.

Within the system, students intact with a variety of animated characters that provide backstories for the math concepts being learned. The main character within the system is called Genie. Genie is a pedagogical agent who encourages students throughout their work in the system. Students are also able to send emails to the Genie and these e-mails are answered in character by Reasoning Mind employees.

2.2 Participants

The data used in this study came from a larger sample of *Foundations* students in the 2016-17 academic year (from August

1, 2016 to June 17, 2017. In the total sample, there were 34,602 students from 462 different schools located in 99 different districts found mainly in Texas. From this larger sample, we selected students who had messaged the Genie and had produced at least 50 words within these messages. We further refined our selection process by sampling students who had completed pre-test and post-test surveys related to math identity, had completed Level A math problems, and were in the 2nd through 5th grades. After the selection criteria, we were left with data from 2,016 students.

2.3 Corpus

The messages sent from the students to the Genie were used as our language sample. One problem with these messages is that many of the samples are quite small consisting of only a few words. Thus, we aggregated all e-mails sent by each student into a single text file in order to create a linguistic representation of each student's language activity. Because the data were extremely noisy with multiple misspellings, non-linguistic garbage, repetitions, and foreign languages, we cleaned the data prior to analysis. First, all non-ASCII characters that could interfere with the NLP tools were removed from the data. Second, all texts were automatically spell-checked and corrected using an implementation of Grammar-Check available in Python. Next, non-English texts were identified and removed and then all non-English words were automatically removed from the data. Lastly, all texts were cleaned of redundancies so that repeated words and phrases were removed.

2.4 Selected NLP Indices

We selected indices from NLP tools that have previously demonstrated strong correlations with math success in previous studies focusing on similar data [6]. The indices and the tools that calculate them are discussed briefly below.

2.4.1 TAALES

The Tool for the Automatic Analysis of Lexical Sophistication (TAALES [13]) is a freely available too that calculates over 150 indices related to basic lexical information, lexical frequency, lexical range, lexical registers, word information features, and psycholinguistic variables. Based on results reported in Crossley et al. [4, 5], we selected indices related to lexical registers that reference the number of registers in which words are found in the Kucera-Francis database (e.g., humor, fiction, and academic registers), psycholinguistic features related to phonological and orthographic neighborhoods (i.e., the number of near neighbors a word has based on its sound or spelling, the frequency of those neighbors, and the Levenshtein distance of the neighbors), the number of senses a word has (i.e., polysemy), the proportion of n-grams in a text that are common in a large reference corpus (i.e., the Corpus of Contemporary American English or the British National Corpus), and word frequency indices (i.e., how frequent a word is in a reference corpus).

2.4.2 TAACO

The Tool for the Automatic Analysis of Cohesion (TAACO [14]) incorporates over 150 classic and recently developed indices related to text cohesion. Based on findings from Crossley et al. [6], we included a single measure of cohesion: Incidence of determiners. Incidence of determiners is related to text givenness with a higher incidence indicating more given information in a text (i.e., a more cohesive text).

2.4.3 TAASSC

The Tool for the Automatic Analysis of Syntactic Sophistication and Complexity (TAASSC [15]) measures large and fined grained clausal and phrasal indices of syntactic complexity and usage-based frequency/contingency indices of syntactic sophistication. Based on Crossley et al. [4, 5], we included a single index related to syntactic complexity: complex T-units. A t-unit is defined as a dominant and any subordinate clause. A complex t-unit is a t-unit the included a dominant and at least one subordinating clause.

2.4.4 SEANCE

The SEntiment ANalysis and Cognition Engine (SEANCE [16]) is a sentiment analysis tool that relies on a number of pre-existing sentiment, social positioning, and cognition dictionaries to measure sentiment, cognition, and social order. From SÉANCE, two indices were selected based on Crossley et al. [4, 5]: quantitative terms and certainty words. Quantitative terms refer to words that assess quantity (e.g., above, addition) while certainty terms refer to words related to sureness and hedging.

2.5 Statistical Analysis

We conducted a number of correlations between the NLP indices described above and the A-level scores for students. We grouped students into three categories based on the number of words produced in their messages to the Genie (50 words and above, 100 words and above, 150 words and above). We further divided students based on grade level (2nd and 3rd grade students and 4th and 5th grade students) because linguistic development occurs in stages that differ based on age [12]. We used the correlations to assess the strength of the relationships between the NLP indices and A-level scores based on the number of words in the text and students' grade levels. Prior to all analyses, we first assessed correlations between text length and A-level scores to ensure length and score were not strongly related.

3. RESULTS

3.1 Text Length and A-Level Problems

We ran initial correlations between text length and success on A-level problems for the three text length categories to examine relationships between length and math success (see Table 1). The correlation indicated that the number of words produced by students in e-mail messages to the Genie had no meaningful correlation with math success within the system.

Table 1

Correlations between text length and A level scores (all grades)

Index	All words (n = 2013)	> 100 words (n = 1105)	> 150 words (n = 684)
Text length	0.012	-0.001	-0.008

3.2 All Grade Levels

We conducted correlations between the selected NLP indices and the three text length categories for students in all grade levels (see Table 2). The correlations indicated that greater text length lead to larger correlation in 12 out of the 13 NLP indices. One NLP index (Bigram Proportion [COCA News]) showed neither an increase or a decrease in correlation. All correlations demonstrated at least a small effect size ($r \geq .100$ [17]) between NLP indices and A-level accuracy scores.

Table 2
Correlations between language features and A level accuracy scores (all grades)

Index	All words (n = 2013)	> 100 words (n = 1105)	> 150 words (n = 684)
Kucera-Francis categories	0.147	0.155	0.184
Phonological neighbors distances (Levenshtein)	0.212	0.209	0.258
Complex T-units	-0.108	-0.134	-0.109
Polysemy (adverbs)	-0.047	-0.098	-0.154
Quantitative terms	0.132	0.124	0.198
Bigram proportion (COCA news)	0.159	0.138	0.154
Phonological Neighbors	-0.219	-0.206	-0.256
Average Levenshtein Distance of closest orthographic neighbors	0.188	0.19	0.208
Trigram proportion (BNC spoken)	0.118	0.138	0.131
Content word frequency (BNC written)	0.138	0.139	0.156
Average frequency of closest orthographic neighbors	-0.192	-0.17	-0.219
Incidence of determiners	0.06	0.058	0.104
Certainty words	0.138	0.092	0.171

3.3 2nd and 3rd Grade Data

We conducted correlations between the selected NLP indices and the three text length categories for students in the 2nd and 3rd grade (see Table 3). The correlations indicated that greater text length yielded larger correlations in 8 out of the 13 NLP indices. Like the full analysis, the NLP index related to ngram proportion scores showed neither an increase or a decrease in correlations. Indices related to register categories, word frequency, and complex T-Units also showed no correlation patterns. All correlations demonstrated at least a small effect size ($r \geq .100$) between NLP indices and A-level accuracy scores. Two indices related to neighborhood effects demonstrated medium effect sizes ($r \geq .300$ [17]).

3.4 4th and 5th Grade Data

We conducted correlations between the selected NLP indices and the three text length categories for students in the 4th and 5th grades (see Table 4). The correlations indicated that greater text length lead to larger correlation in 10 out of the 13 NLP indices. Unlike the 2nd and 3rd grade analysis, three NLP index related to neighborhood effects showed neither an increase or a decrease in correlations. Unlike the 2nd and 3rd grade analysis, indices related lexical sophistication (word frequency and register indices) showe effects for increased text length as did indices related to syntactic complexity (complex T-units). All correlations demonstrated at least a small effect size ($r \geq .100$) between NLP indices and A-level accuracy scores.

Table 3
Correlations between language features and A- level accuracy scores (2nd-3rd grade)

Index	All words (n = 1046)	> 100 words (n = 558)	> 150 words (n = 333)
Kucera-Francis categories	0.159	0.158	0.159
Phonological neighbors distances (Levenshtein)	0.225	0.247	0.323
Complex T-units	-0.079	-0.127	-0.036
Polysemy (adverbs)	-0.014	-0.101	-0.126
Quantitative terms	0.116	0.132	0.217
Bigram proportion (COCA news)	0.165	0.131	0.163
Phonological Neighbors	-0.222	-0.256	-0.322
Average Levenshtein Distance of closest orthographic neighbors	0.203	0.24	0.277
Trigram proportion (BNC spoken)	0.125	0.11	0.116
Content word frequency (BNC written)	0.169	0.138	0.149
Average frequency of closest orthographic neighbors	-0.203	-0.257	-0.299
Incidence of determiners	0.037	0.043	0.096
Certainty words	0.146	0.123	0.196

Table 4
Correlations between language features and a accuracy scores (4th-5th grade)

Index	All words (n = 967)	> 100 words (n = 547)	> 150 words (n = 351)
Kucera-Francis categories	0.078	0.128	0.198
Phonological neighbor distances (Levenshtein)	0.138	0.142	0.172
Complex T-units	-0.148	-0.14	-0.197
Polysemy (adverbs)	-0.084	-0.083	-0.173
Quantitative terms	0.106	0.091	0.152
Bigram proportion (COCA news)	0.084	0.109	0.103
Phonological Neighbors	-0.167	-0.126	-0.169
Average Levenshtein Distance of closest orthographic neighbors	0.118	0.111	0.113
Trigram proportion (BNC spoken)	0.071	0.14	0.111
Content word frequency (BNC written)	0.066	0.128	0.153
Average frequency of closest orthographic neighbors	-0.137	-0.044	-0.105
Incidence of determiners	0.049	0.06	0.105
Certainty words	0.068	0.021	0.108

4. DISCUSSION/CONCLUSION

The key finding from this study is that longer texts lead to stronger associations between linguistic features reported by NLP tools and math success within an online tutoring system. importantly, text length was not correlated with math scores (i.e., students who wrote longer emails within the system did not have lower or higher math scores). However, in almost all cases, increased text lengths led to stronger correlations. This was especially true for the correlations that included all grade levels in which 12 of 13 indices showed increased correlations. Importantly, for all grades, longer text length led to correlations that demonstrated at least small effect sizes ($r > .010$).

Deconstructing the grade level analyses also leads to interesting comparisons. Chief among these is the finding that phonological neighborhood effects have stronger effects for 2nd and 3rd graders when compared to 4th and 5th graders. This is likely the result of younger students demonstrating greater development in word learning than older students as a result of language acquisition stages [12]. In contrast, older students generally develop stronger syntactic skills once lexical development stabilizes [49], which may be reflected in the stronger correlations we see between our single syntactic complexity index (complex T-units) and math scores for 4th and 5th graders. This notion gains support when we compare it to the lower correlations reported between math scores and complex t-units for 2nd and 3rd graders.

While the findings of this study provide evidence of both text length and grade level differences in NLP analyses, they need to be tested on different data sets. For instance, the data used in this analysis included all available data regardless of text length. This meant that the corpus containing texts greater than 50 words also contained texts that had more than 100 and more than 150 words. Future studies should examine bands of texts that contain only specific text lengths (e.g., a band consisting of only texts with lengths between 50 and 99 words compared to texts consisting of 100 to 149 words). Future studies should also see whether the results reported here extend to other learning domains (e.g., literacy or science domains), other text types (e.g., questions, summaries, self-explanations), other NLP tools and features, and other grade or age levels.

5. ACKNOWLEDGMENTS

The author is deeply indebted to Victor Kostyuk, Laura Allen, Matthew Labrum, and Jaclyn Ocumpaugh for their help with this manuscript. This research was supported in part by the National Science Foundation (DRL- 1418378). Ideas expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

6. REFERENCES

[1] CROSSLEY, S. A., ALLEN, L. K., AND MCNAMARA, D. S. 2016. Writing Pal: A writing strategy tutor. In *Adaptive Educational Technologies for Literacy Instruction,* S. A. Crossley and D. S. McNamara, Eds. Routledge, New York, 204-224.

[2] JACKSON, G.T., GUESS, R.H., AND MCNAMARA, D.S. 2010. Assessing cognitively complex strategy use in an untrained domain. *Topics in Cognitive Science 2* (Dec), 127-137.

[3] CROSSLEY, S. A., PAQUETTE, L., DASCALU, M., MCNAMARA, D., AND BAKER, R. 2016. Combining click-stream data with NLP tools to better understand MOOC completion. In *Proceedings of the 6th International Learning Analytics and Knowledge Conference.* AMC, NY, 6-14.

[4] CROSSLEY, S. A., AND KOSTYUK, V. 2017. Letting the Genie out of the Lamp: Using Natural Language Processing tools to Predict math performance. In *Language, Data, and Knowledge.* LDK'17. Lecture Notes in Computer Science, Vol 10318. Springer, Cham, Switzerland, 330-342.

[5] CROSSLEY, S. A., OCUMPAUGH, J., LABRUM, M., BRADFIELD, F., DASCALU, M., & BAKER, R. (in press). Modeling Math Identity and Math Success through Sentiment Analysis and Linguistic Features. *Proceedings of the 10th International Conference on Educational Data Mining (EDM).*

[6] VARNER, L. K., JACKSON, G. T., SNOW, E. L., AND MCNAMARA, D. S. 2013. Does size matter? Investigating user input at a larger bandwidth. In *Proceedings of the 26th International Florida Artificial Intelligence Research Society Conference*. FLAIRS'13. AAAI Press, Menlo Park, CA, 546-549.

[7] ALLEN, L. K., SNOW, E. L., AND MCNAMARA, D. S. 2015. Are you reading my mind? Modeling students' reading comprehension skills with Natural Language Processing techniques. In *Proceedings of the 5th International Learning Analytics & Knowledge Conference.* LAK'15. ACM, Poughkeepsie, NY, 246-254

[8] KOPP, K. J., JOHNSON, A. M., CROSSLEY, S. A, AND MCNAMARA, D. S. 2017. Assessing question quality using Natural Language Processing. *Proceedings of the 18th International Conference on Artificial Intelligence in Education.* 523-527.

[9] ZHANG, T., HUANG, M., AND ZHAO, L. 2018. Learning structured representation for text classification via reinforcement learning. *Association for the Advancement of Artificial Intelligence*.

[10] VYAS, V. AND UMA, V. 2018. An extensive study of sentiment analysis tools and binary classification of tweets using rapid miner. *Procedia Computer Science* 125, 329-335.

[11] YOUNG, T., HAZARIKA, D., PORIA, S., AND CAMBRIA, E. 2017. Recent trends in deep learning based Natural Language Processing. arXiv:1708.02709v4

[12] FROMKIN, V. 1983. *An Introduction to Language*. Third Edition. New York. CBS College Publishing.

[13] KYLE, K., CROSSLEY, S. A., AND BERGER, C. 2017. The Tool for the Automatic Analysis of Lexical Sophistication (TAALES): Version 2.0. *Behavior Research Methods,* 1-17.

[14] CROSSLEY, S. A., KYLE, K., AND MCNAMARA, D. S. 2016. The Tool for the Automatic Analysis of Text Cohesion (TAACO): Automatic assessment of local, global, and text cohesion. *Behavior Research Methods 48,* 4, 1227-1237.

[15] KYLE, K. AND CROSSLEY, S. A. 2018. Measuring syntactic complexity in l2 writing using fine-grained clausal and phrasal indices. *Modern Language Journal. doi:10.1111/modl.12468*

[16] CROSSLEY, S. A., KYLE, K., AND MCNAMARA, D. S. 2017. Sentiment Analysis and Social Cognition Engine (SEANCE): An automatic tool for sentiment, social cognition, and social order analysis. *Behavior Research Methods 49,* 3, 803-821.

[17] COHEH, J. 1992. A power primer. *Psychological Bulletin, 112,* 1, 155-159.

Sensor-Free Predictive Models of Affect in an Online Learning Environment

Avery Harrison
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA
aeharrison@wpi.edu

Naomi Wixon
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA
nbwixon@wpi.edu

Anthony Botelho
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA
abotelho@wpi.edu

Ivon Arroyo
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA
iarroyo@wpi.edu

ABSTRACT

A significant amount of research has illustrated the impact of student emotional and affective state on learning outcomes. Just as human teachers and tutors often adapt instruction to accommodate changes in student affect, the ability for computer-based systems to similarly become affect-aware, detecting and personalizing instruction in response to student affective state, could significantly improve student learning. Personalized and affective interventions in tutoring systems can be realized through affect-aware learning technologies to deter students from practicing poor learning behaviors in response to negative affective states and to optimize the amount of learning that occurs over time. In this paper, we build off previous work in affect detection within intelligent tutoring systems (ITS) by applying two methodologies to develop sensor-free models of student affect with only data recorded from middle-school students interacting with an ITS. We develop models of four affective states to evaluate and determine significant predictors of affect. Namely, we develop a model which discerns students' reported interest significantly better than majority class.

Keywords

Intelligent tutoring systems; Affect detection.

1. INTRODUCTION

The ability to identify and deliver personalized interventions that are effective for individual students can greatly benefit the learning process by recognizing and addressing specific student needs. However, it's often unfeasible to realize personalized instruction and support in traditional classrooms with large numbers of students per teacher. With growing access to technology in classrooms, online learning platforms such as MathSpring have provided personalized learning opportunities that have shown positive achievement outcomes for student users [3]. Recently, such work has shifted focus to acknowledge and leverage the impact that emotion has on learning. Affect-aware learning technologies, including online learning platforms, can be developed and deployed to monitor and predict affect to provide appropriate interventions to maximize student learning.

Modeled after the control-value theory of emotion in education [16] and previous work in affect detection, we aim to develop sensor-free predictive models of affect from user behavior and performance within MathSpring. We will use students' self-reported levels of confidence, interest, excitement, and frustration during four user sessions to create predictive models and detect how student behaviors in the system relate to levels of affective states. Doing so will further efforts to build, study, and deploy affective interventions within MathSpring to optimize learning with feasible means for educational settings.

2. LITERATURE REVIEW

2.1 Affect and Control-Value Theory

A growing body of research has investigated emotion and affect in the context of education [12, 13]. To differentiate emotions and affect, consider emotions to be intuitive feelings, such as joy and anger, while affect broadly captures the manifestations of those feelings, such as pleasure and frustration, particularly in educational settings [17]. From perspectives in psychology, education, and computer science, a large amount of evidence suggests that student affect influences learning and deeper comprehension, both positively and negatively [5, 10, 12]. This research highlights the importance of affect in learning to provide content that effectively challenges students.

Our framework is based on the control-value (CV) theory of emotion [16]. Pekrun's CV theory of achievement emotions posits that student beliefs of their control over success in a subject and their value in understanding said subject will most influence their affect and, consequently, overall learning. For example, a student might feel enjoyment during an activity for which that student feels greater confidence in learning the content. The CV theory attributes student affect to feelings of control and subject value within a learning environment, underscoring the necessity of providing students with appropriately challenging tasks and adaptive content to maintain emotions that will positively influence learning in a given activity.

2.2 Sensor-Free Affect Detection

Efforts have been made to develop sensor-free affect detectors with tutoring systems for educational settings, particularly by pairing student log files with human observations to detect behavior that might be representative of affect. For instance, Baker and colleagues developed BROMP [15] for observers to code student affect over short intervals and then match the observed affect with student activity logs [8]. Researchers have also observed facial expressions and body movement to create a framework for mapping affect onto student behavior [10, 17]. However, it is difficult to implement student affect detectors with physiological sensors or observational data in a permanent school setting [7] due to cost and the potential threat to validity introduced by such methods caused by alterations of the learning environment. Researchers have previously tried to predict self-reported affect with log data and questionnaires [9] but less work has been done with solely log data.

3. CURRENT STUDY

MathSpring is an intelligent tutoring system that covers Common Core mathematics curriculum for students in 6th-10th grade to prepare for standardized tests [1]. The system adapts to provide content that will likely keep the student in the zone of proximal development [3] while providing scaffolding and fostering growth mindsets through personalized, pedagogical support. Affective support is realized through text, audio, and images from an animated learning companion as students solve problems [3]. Studies have found that using MathSpring leads to significant performance gains on standardized math tests as opposed to students who do not practice with MathSpring [1].

We currently aim to utilize student data from MathSpring build affect predictors from only student logs. We intend to respectively construct predictive models for confidence, interest, excitement, and frustration levels reported over brief intervals in user sessions. Based on previous affect detection work, and gaps in the literature, we hypothesize that students changing topics and viewing progress in MathSpring will contribute to affect predictions [1]. We predict that topic changes may indicate frustration and be negatively related to positive affective states.

This study was conducted with 85 eighth-grade students at a middle school in Massachusetts. Between December 2016 and May 2017, students participated in four, hour-long sessions with MathSpring. In each session, students worked on assigned problem sets corresponding with class material. Students typically completed the assigned problem set in that time or completed the set in the next session. Throughout each session, users saw a learning companion that delivered messages to remind students of the hint button or provide encouragement. Previous work has looked at the effects of interventions with different affective messages, such as empathetic, growth mindset, and success/failure messages [1]. Growth mindset messages were used in this study because they are the default for MathSpring. If a user selected an incorrect answer, the hint button would flash. After a second wrong attempt, the learning companion delivered a growth mindset message. Students could skip problems or return to the "My Progress" page any time where they could view topic mastery and choose to continue, change topics, challenge themselves, or review content.

Drawing from previous work on affect detection in learning technologies [1, 2, 4, 6], we inquired about levels of excitement, interest, confidence, and frustration during user sessions. Roughly every five minutes between problems, students received a prompt to self-report affect on a 5-point Likert scale ranging from 1=*Not at all* to 5=*Extremely,* with the option to skip the self-report. Prompts randomly alternated between the affective states but contained the same wording. For example, the prompt for Confidence would read, "*Please tell us how you are feeling. Based on the last few problems tell us about your level of Confidence in solving math problems*."

4. MODELS AND ANALYSES

We first reconstructed data from student log files. Affect self-reports were randomized throughout the user sessions so the order and summation of self-reports for each affect varied by user. For example, a student could have reported on confidence followed by interest level while another student could have been prompted for frustration and then excitement level. Due to this variation between and within students, we chose to use the "mini-sessions" of activity between each affect report. This is supported by previous findings that recently completed problems are more predictive of affect than an entire user session [6] and alleviates the possible effect of elapsed time on affect reports.

Table 1. Descriptive statistics on affect self-reports.

	Excitement	Frustration	Interest	Confidence
N Affect Reports	138	129	133	154
Mean (*SD*)	1.78 (1.17)	2.36 (1.67)	1.98 (1.29)	3.23 (1.53)

With mini-sessions of self-reported affect (*N*=554; Table 1), we aggregated behavior variables, such as the number of problems seen between reports, that corresponded with a given affect report then separated mini-sessions by affect. Observations without a reported emotion level (*N*=196) were culled. A PCA with "mini-session" level variables revealed four factors. We selected one variable per factor for the models. *Topic changes* refer to a student changing problem sets due to completion or topic mastery, prolonged poor performance, or self-electing to return to the progress page and choose a different topic. The *average number of hints* refers to the average seen per problem. The *percentage of problems answered correctly* is calculated within the "mini-session". Lastly, the *number of interventions* sums hint button flashes and messages from the learning companion during problem solving.

Based on past MathSpring work [1], we tried two methods of building predictive models of affect by constructing logistic regressions with five-fold cross-validations at the student level. The "at least somewhat" models attempt to predict whether students would report "Not at all" to "A little" (1-2 on the self-report scale) or "Somewhat" to "Extremely" (3-5) of a given affect. Then, the "at least a little" models attempt to predict whether students reported any degree of a given affect (2-5) or not at all (1).

Table 2 summarizes the performance of models. Notably, both models of interest perform comparably to other predictive models of affect (kappa > 0.20) [14]. While there is variation across affect and discretization, with both Confidence models and the "at least a little" model for Frustration performing below chance (AUC < 0.50; kappa < 0), five of the models appeared to be performing above chance with disagreeing AUC and kappa values. Unlike AUC, accuracy, F_1, and kappa values are sensitive to the choice of rounding threshold of model estimates, particularly with unbalanced labels. This incongruence between AUC and kappa has been seen in other work on sensor-free affect detection using deep learning [8]. Given the imbalance of labels within each affective state, we calculated an optimized

Table 2. Logistic regression model performance.

Model	AUC	Kappa	F_1	Optimized Accuracy	Optimized F_1	Optimized Kappa
At Least Somewhat						
Interest	**0.75**	**0.24**	**42.41**	**72.29**	**58.54**	**0.38**
Confidence	**0.70**	**0.02**	**73.76**	**68.41**	**70.33**	**0.32**
Excitement	**0.68**	**0.10**	**25.00**	**63.17†**	**37.50**	**0.10**
Frustration	0.53	-0.04	18.46	59.36	18.46	-0.04
At Least A Little						
Interest	**0.73**	**0.32**	**60.34**	**67.14***	**61.57**	**0.35**
Confidence	0.69	-0.04	84.18	64.76†	70.94	0.22
Excitement	**0.62**	**0.06**	**42.11**	**65.42**	**58.18**	**0.20**
Frustration	0.39	-0.17	32.03	36.33†	32.03	-0.17

Note: Bolded rows indicate model performance above chance ($0.50 < \text{AUC} \leq 1$; $0 < \text{kappa} \leq 1$). Optimized accuracies significantly better ($p<.05$) than a base rate model are denoted with (*), while optimized accuracies significantly worse than base rate are denoted with (†).

metric by learning a reasonable rounding threshold of model estimates using the training set of each fold. We also compared each model's optimized accuracy to the respective base rate, majority class model to determine significance. It is found that only the model for "at least a little" Interest has a significantly higher accuracy than the base rate. Table 3 details standardized coefficients for each model. Number of interventions was the most frequent predictor across affects and discretization levels. Percentage of correct problems was also a strong predictor of interest ($p < 0.01$). Topic changes positively predicted interest and excitement and negatively predicted frustration.

Table 3. Standardized coefficients (β) of predictors by model.

Model	Topic Changes	Avg. Hints	Correct Problems (%)	Number of Interventions
At Least Somewhat				
Interest	**0.69**	-0.10	**0.95**	**-0.80**
Confidence	0.39	0.40	0.04	**-0.48**
Excitement	0.48	0.12	-0.23	**-0.80**
Frustration	**-0.83**	0.17	-0.37	**0.75**
At Least A Little				
Interest	0.41	-0.13	**0.67**	**-0.53**
Confidence	0.37	0.16	0.01	**-0.56**
Excitement	**0.55**	-0.03	0.10	**-1.04**
Frustration	-0.26	0.10	0.05	0.26

5. DISCUSSION

We presented predictive models of affect within MathSpring with a model of "at least a little" interest that performs significantly well. In general, the "at least somewhat" models perform better, suggesting that this discretization split should be used in future projects to predict student affect. While some of the models do not perform well, this is not surprising given that sensor-free affective models are more difficult to build than models profiting from detectors or pre- and post-study data.

However, it is surprising that the number of topic changes, contrary to our hypothesis, was positively related to interest and excitement levels and negatively related to frustration. This implies that higher frequencies of topic changes between affect reports indicate positive affective states. Conversely, a student who does not change topics between affect reports is more likely to report a higher level of frustration. We assumed that students would change topics if they performed poorly (indicating that the content is too challenging to be productive) or were bored. However, students could also change topics if they mastered or completed a topic (indicating the content is too easy). Considering the positive relationship between interest and topic change, and excitement and topic change, perhaps students were more likely to change topics because of completion or mastery. This suggests that students might conflate the concepts of interest and excitement with feelings of achievement.

The other predictor to note, number of interventions, was the most common, statistically significant predictor of affect level across models. Number of interventions was negatively related to positive affect which suggests that fewer interventions led to higher reports of positive affective states. Conversely, the number of interventions positively predicted frustration, suggesting that more interventions predicted a higher level of frustration. Assuming that interventions increased as student attempts increased, it is unsurprising that higher numbers of interventions precede higher reports of frustration and lower reports of positive affective states. The number of topic changes and interventions between affect reports were the main predictors of affect across models, while percent of problems answered correctly only positively related to interest. The lack of strength in the four predictive attributes suggests that we should consider other variables from the four PCA components.

There are other caveats to consider. Namely, self-report from middle school students might not be accurate and prompting students to self-report throughout user sessions might disrupt natural affect. Also, the type of intervention might influence affect rather than the quantity of interventions. For instance, students who saw affirmative messages after answering a problem correctly might have felt differently towards the learning companion and MathSpring than a student who saw growth mindset messages after attempting a problem multiple times. That said, using interventions as a variable in the sensor-free predictive models is only beneficial to data from MathSpring until we better comprehend the underpinnings of how interventions influence student affect more broadly.

This work poses future directions to give better consideration to these questions. It might be worthwhile to construct ordinal regression to predict the level of affect reported rather than a binary classification. We also intend to create a feature that indicates the previous self-report level of the affect in question. This feature was not included for the initial round of analyses due to the randomization of affect report ordering. A student might only report on a given affect once or twice towards the beginning of the session, rendering the information less useful than if the same affect were reported on twice in a row across a shorter span. Even with potential irregularity of affect reporting, previously-reported same-affect level could be suggestive of the dynamics of affect throughout user sessions. Pursuing these directions will help us better understand the dynamic between student affect and behavior in tutoring systems.

6. CONCLUSION

We presented a high-performing predictive model of interest, as well as predictive models of excitement and confidence that perform above chance, demonstrating the ability to build sensor-free detectors of affect in MathSpring. Given the limitations of the current models and future plans with the data from this study, we consider this to be a first effort. We intend to utilize the data to improve sensor-free affect detection so that socio-emotional interventions in MathSpring can be better realized to optimize student support and learning. Progress in sensor-free affect detection research has positive implications for classroom implementation of affect-aware learning technologies and sustainable data collection through student activity files.

7. REFERENCES

[1] Arroyo, I., Wixon, N., Muldner, K., Karumbaiah, S., Lizarralde, R., Allessio, D., Burleson, W., and Woolf, B. P. 2017. *Addressing student emotion in personalized digital learning environments*. Manuscript submitted for publication.

[2] Arroyo, I., Shanabrook, D. H., Burleson, W., & Woolf, B. P. 2012. Analyzing affective constructs: Emotions ‘n attitudes. *Proceedings of the 11th International Conference on Intelligent Tutoring Systems, Greece*, 714-715.

[3] Arroyo, I., Woolf, B. P., Burleson, W., Muldner, K., Rai, D., and Tai, M. 2014. A multimedia adaptive tutoring system for mathematics that addresses cognition, metacognition, and affect. *International Journal on Artificial Intelligence in Education, Special Issue on “Landmark AIED Systems for STEM Learning”,* 24, 387-426.

[4] Arroyo, I. 2014. Analyzing affective constructs: Emotions, attitudes, and motivation. Retrieved from: http://digitalcommons.wpi.edu/ssps-papers/2

[5] Arroyo, I., Cooper, D.G., Burleson, W., and Woolf, B.P. 2010a. Bayesian networks and linear regression models of students’ goals, moods, and emotions. In *Handbook of Educational Data Mining,* C. Romero, S. Ventura, M. Pechenizkiy, and R. Baker (Eds.). CRC Press, Boca Raton, 323-338.

[6] Baker, R.S.J.d., 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 (2010), 223-241.

[7] Baker, R.D., Gowda, S., 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. *Proceedings of the 5th International Conference on Educational Data Mining*, 126-133.

[8] Botelho, A., Baker, R. S., and Heffernan, N. T. 2017. Improving sensor-free affect detection using deep learning. In *International Conference on Artificial Intelligence in Education*, 40-51, Springer, Cham.

[9] Conati, C., and Maclaren, H. 2009. Empirically building and evaluating a probabilistic model of user affect. *User Modeling and User-Adapted Interaction*, 19, 267-303.

[10] D’Mello, S.K., and Graesser, A.C. 2010. Multimodal semi-automated affect detection from conversational cures, gross body language, and facial features. *User Modeling and User-Adapted Interaction*, 20, 2, 147-187.

[11] D’Mello, S., Person, N., and Lehman, B. 2009. Antecedent-consequent relationships and cyclical patterns between affective states and problem solving outcomes. In *Artificial Intelligence in Education. Building Learning Systems that Care: from Knowledge Representation to Affective Modelling,* V. Dimitrova, R. Mizoguchi, B. du Boulay, and A. Grasser (Eds.). IOS Press, 57-64.

[12] D’Mello, S.K., Strain, A.C., Olney, A., and Graesser, A. 2013. Affect, meta-affect, and affect regulation during complex learning. In *International Handbook of Metacognition and Learning Technologies*, R. Azevedo and V. Aleven (Eds.). Springer International Handbooks of Education 26.

[13] Ekman, P. 1999. Basic emotions. In *Handbook of Cognition and Emotion,* T. Dalgleish and M. Power (Eds.). John Wiley & Sons Ltd., 45-60.

[14] 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.

[15] Ocumpaugh, J., Baker, R.S., and Rodrigo, M.M.T. 2015. Baker Rodrigo Ocumpaugh Monitoring Protocol (BROMP) 2.0 Technical and Training Manual. *New York, NY and Manila, Philippines: Teachers College, Columbia University and Ateneo Laboratory for the Learning Sciences.*

[16] Pekrun, R. 2006. The control-value theory of achievement emotions: Assumptions, corollaries, and implications for educational research and practice. *Educational Psychology Review*, 18, 315-341.

[17] Woolf, B., Burleson, W., Arroyo, I., Dragon, T., Cooper, D., and Picard, R. 2009. Affect-aware tutors: Recognising and responding to student affect. *International Journal of Learning Technology*, 4, 3/4 (2009), 129-164.