JOURNAL OF LEARNING ANALYTICS

(2014). Constructing Programming Tests from an Item Pool: Pushing the Limits of Student Knowledge using Assessment and Learning Analytics. *Journal of Learning Analytics*, *1*(3), 161–164.

Constructing Programming Tests from an Item Pool: Pushing the Limits of Student Knowledge using Assessment and Learning Analytics

Vladimir Ivančević
University of Novi Sad, Faculty of Technical Sciences, Serbia
dragoman@uns.ac.rs

ABSTRACT: Tests targeting the upper limits of student ability could aid students in their learning. This article gives an overview of an approach to the construction of such tests in programming, together with ideas on how to implement and refine them within a learning management system.

KEYWORDS: Assessment creation, item difficulty, student competence, programming, LMS

1 INTRODUCTION

In today's IT-dominated world, there is great demand for courses in programming. Consequently, teachers employ computer-based assessment to manage larger groups of students better. Programming assignments and tests are commonly used in programming courses. Grading a programming assignment is usually a manual activity, but there are attempts to automate the grading process (Singh, Gulwani, & Solar-Lezama, 2013). On the other hand, both test grading and construction may be readily automated.

More information about learning may be extracted in test matching the upper limits of ability, as opposed to tests covering only the areas in which the student is fully competent or incompetent. Such tests could be used to evaluate student ability and help students discover gaps in their knowledge. It is important not to discourage students with tests too far from their competence, as psychological aspects of test taking, such as effort and boredom, may skew the results (Asseburg & Frey, 2013).

The main objective of the study is to offer an approach to the construction of tests in programming courses that could be utilized in modern learning environments. Tests created through this approach would match the ability of the assessed group of students and cover the required concepts. The paper presents the already established structure of the approach (Ivančević, Knežević, Pušić, & Luković, 2014) and the plan on how to refine the approach by implementing it in the context of a learning management system (LMS).

(2014). Constructing Programming Tests from an Item Pool: Pushing the Limits of Student Knowledge using Assessment and Learning Analytics. *Journal of Learning Analytics*, *1*(3), 161–164.

2 MOTIVATION AND RELATED WORK

Using the Item Response Theory (IRT) (Baker, 2001), fully adaptive assessment may be performed. However, the application of IRT is often not straightforward. Most IRT models are unidimensional, i.e., they acknowledge only a single trait (ability). This might be impractical for programming tests as items (questions) may target a broad set of skills: from rote learning over the mastery of a single programming concept to intense concentration during the analysis of a tricky code snippet. Moreover, these models need to be calibrated, which, depending on the chosen model, may require at least 500 or even 1,000 examinees (Hulin, Lissak, & Drasgow, 1982). For many programming courses, this could mean numerous years of sample collection and having to deal with the issue of item parameter drift (Veerkamp & Glas, 2000).

In response to the aforementioned problems, in the proposed approach, item difficulty is estimated using the proportion of correct answers, which seems to be a good alternative to the IRT measures because of its strong relation to the true difficulty parameter values (Wauters, Desmet, & van den Noortgate, 2011). This information is also easily obtained in practice and may be readily utilized to circumvent the demanding IRT calibration. For the estimation of new or rarely utilized items, there is little or no actual assessment data available, so instead of the typical measures, a new difficulty estimation technique was utilized.

3 APPROACH TO TEST CONSTRUCTION

The overall structure of the approach was defined in Ivančević et al. (2014), while the initial evaluation was conducted in the context of an introductory programming course and an in-house computer-based assessment system at the Faculty of Technical Sciences (FTS), University of Novi Sad (Serbia). Tests generated through this approach are expected to match the set difficulty level (or the estimated ability level of a student) and cover the selected knowledge concepts. In order to achieve this, there needs to be an item pool, where each item is annotated with a difficulty estimate and a set of knowledge concepts that it covers. Moreover, for each student, there should be an estimate of the ability level. Once all these pieces of information are available, a genetic algorithm uses them as a basis in the process of assembling a set of items that match the desired difficulty and concept coverage.

In order to enrich the item pool originally used in the existing assessment system, two activities were conducted. First, a concept map of all relevant concepts concerning structured programming and the C programming language was created and each item from the pool was related to a subset of these concepts. Second, each item was ranked and classified by its difficulty; that is, calculated utilizing either the percentage of correct answers, if there are historical assessment records, or the complexity of the item estimated by a support vector machine classifier, if there are few or no previous records available. On the other hand, the overall competence level of each student is estimated using a regression model

(2014). Constructing Programming Tests from an Item Pool: Pushing the Limits of Student Knowledge using Assessment and Learning Analytics. *Journal of Learning Analytics*, *1*(3), 161–164.

that features previous test scores. The quality of the generated tests was verified for scenarios concerning assessments of varying difficulty and scope.

4 APPROACH REFINEMENT WITH LMS

The plan for the next phase of the research includes implementing the proposed approach in an open source LMS. There are two main reasons for the transition from the current solution (accessible only from the laboratory) to a constantly available web-based system. First, this would allow students to take ever-changing quizzes, check their learning progress, practice more often, and better prepare for their pre-exam assessments in the regular computer classes. Currently, Canvas LMS is used in several programming courses at FTS. However, instructors tend to create several quizzes in the LMS, which usually are tried out by students quickly and only serve as a narrow set of examples of the actual classroom assessment. If tests are generated within the LMS, students could use quizzes to learn from their mistakes and discuss important points with their peers and instructors within the same environment.

Second, data concerning student behaviour and performance in such quizzes could be used to improve the estimates of item difficulty and student knowledge level. By releasing the enriched item pool to the LMS, a larger number of data records about items could be collected, which in turn should lead to more reliable difficulty estimates and, consequently, quizzes better matching the true competence level of the engaged student. For each item, logs could be used as a source of information about the percentage of correct answers, as well as the actual answer times. With these larger data sets, the links between item difficulty and item attributes, such as concept coverage, length, and keyword count, could be investigated using correlation analysis or feature selection algorithms. This could uncover difficulty-related patterns that may be useful in the estimation of the difficulty of completely new test items.

5 CONCLUSION

Once fully implemented in a LMS, the devised approach is expected to yield assessments that promote student learning. With respect to their difficulty, the items automatically selected for inclusion in a test are focused on the current top abilities of target students. This in turn should help students learn which areas they know well, which need further effort, and which are still beyond their reach.

ACKNOWLEDGEMENTS

The research presented in this paper was supported by the Ministry of Education, Science, and Technological Development of the Republic of Serbia under Grant III-44010.

(2014). Constructing Programming Tests from an Item Pool: Pushing the Limits of Student Knowledge using Assessment and Learning Analytics. *Journal of Learning Analytics*, *1*(3), 161–164.

REFERENCES

Asseburg, R., & Frey, A. (2013). Too hard, too easy, or just right? The relationship between effort or boredom and ability–difficulty fit. *Psychological Test and Assessment Modeling*, *55*(1), 92–104.

Baker, F. B. (2001). *The basics of item response theory* (2nd ed.). College Park, MD: ERIC Clearinghouse on Assessment and Evaluation. Available from http://eric.ed.gov/?id=ED458219

Hulin, C. L., Lissak, R. I., & Drasgow, F. (1982). Recovery of two- and three-parameter logistic item characteristic curves: A Monte Carlo study. *Applied Psychological Measurement*, *6*(3), 249–260.

Ivančević, V., Knežević, M., Pušić, B., & Luković, I. (2014). Adaptive testing in programming courses based on educational data mining techniques. In A. Peña-Ayala (Ed.), *Educational data mining* (pp. 257–287). Cham, Switzerland: Springer International Publishing.

Singh, R., Gulwani, S., & Solar-Lezama, A. (2013). Automated feedback generation for introductory programming assignments. *Proceedings of the 34th ACM SIGPLAN Conference on Programming Language Design and Implementation* (PLDI ’13), 16–22 June 2013, Seattle, WA, USA (pp. 15–26). New York: ACM.

Veerkamp, W. J. J., & Glas C. A. W. (2000). Detection of known items in adaptive testing with a statistical quality control method. *Journal of Educational and Behavioral Statistics*, *25*(4), 373–389.

Wauters, K., Desmet, P., & van den Noortgate, W. (2011). Acquiring item difficulty estimates: A collaborative effort of data and judgment. *Proceedings of the 4th International Conference on Educational Data Mining* (EDM ’11), 6–8 July 2011, Eindhoven, Netherlands (pp. 121–127). International Educational Data Mining Society.