Implementation of Elo-Rating method in Recommending Coding Exercises to Programming Students

Katerina Mangaroska, Özlem Özgöbek, Boban Vesin, Michail Giannakos
Department of Computer Science
Norwegian University of Science and Technology
Norway

It is hypothesized that adaptive learning systems will revolutionize the learning process by offering content and resources that match learner's skills and needs. The main purpose of adaptive e-learning systems is to apply personalization and adaptation in the learning process through tailored resources and assessment that could aid the dynamics of the learning and teaching processes. Adaptive systems consider the strengths and weaknesses of every learner, an information that is included and continuously updated in meaningful learner models. Moreover, the degree of adaptiveness can be extended from a basic predefined adaptive assignments based on difficulty to autonomous intelligent tutoring systems that continuously adapt on learner's skills, needs, and performance. Different adaptive assessments methods within programming courses are mostly related to item response theory (Vega, Bolaños, Nieto, & Baldiris, 2012), different variants of Elo rating system (Pelánek, Papoušek, Řihák, Stanislav, & Nižnan, 2017) or Bayesian networks (Hooshyar et al., 2016). Consequently, to achieve effective adaptive learning, it is necessary to address several interrelated issues, particularly the assessment of students' abilities, building of learner model, educational content difficulty estimation, and implementation of adaptive features in educational systems.

A group of researchers at the Norwegian University of Science and Technology (NTNU) aim to implement an Elo-rating system in ProTuS, an Intelligent Tutoring System, for an introductory programming course for first year computer science students. The presented work-in-progress focuses on adaptive assessment and adaptation of the coding exercises assigned to the students. A prerequisite for adaptive choice of recommended exercises, in which the difficulty of the task is constantly aligned to the knowledge level of the user, is to have items with a known difficulty level (Wauters, Desmet, & Van Den Noortgate, 2012). Several researchers proposed methods to automate assessment of learner's programming assignments (Pieterse, 2013; Trætteberg & Aalberg, 2006). However, those methods rarely consider the personal characteristics of the users to carry a truly adaptive assessment. Thus, for the purpose of this study the authors implemented adapted Elo-rating method, to estimate the knowledge rating of users and the difficulty of the provided coding exercises, that generates recommendations of assignments to students based on their current knowledge level.

The approach adopted in this study is based on System Development research (Nunamaker Jr, Chen, & Purdin, 1990) and the best practices in User-centered design, allowing researchers to address learning and design problems in real-world settings, with two primary goals: to construct knowledge and develop solutions (McKenney & Reeves, 2013). The study was carried out in the spring semester 2018 at NTNU in a course of Introduction to Java programming. Before interacting with ProTuS, the students were introduced with NTNU policy for ethical and data privacy issues, as well as the purpose of the study and their voluntarily participation. The smart interactive learning content in ProTuS, consist of four type of activities that students could engage with, but our study focused only on the coding exercises. Each coding exercise has a problem description and a baseline code. When students submit the code, the code is being tested against a set of unit tests for that problem and the user receives a feedback whether the tests were passed or not. The implemented Elo-rating method for the user and the content is represented by a number which increases or decreases depending on the number of successful or failed attempts to solve a coding exercise. The core of the system's recommendation process lies in ranking learners' knowledge and recommending coding exercises that match their current knowledge. While browsing the content, the learner is recommended with 5 coding exercises that most suit learner's knowledge level (Figure 1). Thus the learner has opportunity to choose between recommended exercises or to try solving more complicated (that could possible earn him/her more ranking points) or less complicated (that will potentially be less challenging but, at the same time, earn him/her less ranking points).

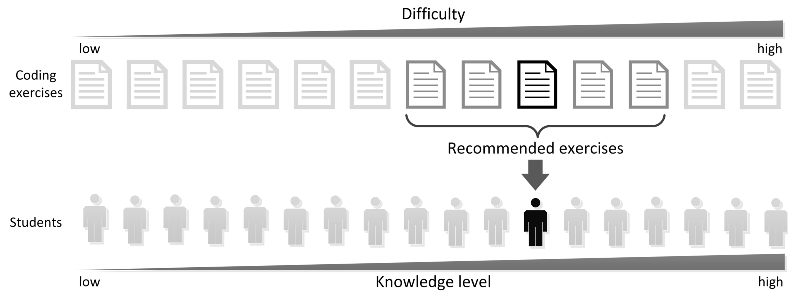

Figure 1. The recommendation of coding exercises

The proposed method estimates the probability that a user i is able to solve the coding exercise j based on its current rank and the difficulty of the coding exercise j. W represents the result from solving the assigned exercise (i.e. the success rate) in the range [0,1], calculated by the following formulas:

$$W = \frac{As + Ao}{2xAo}$$

where As represents the number of successful attempts, and Ao represents the number of overall attempts. This method presented in the paper differs from other implementations of the same algorithm which calculates the success rate W. The proposed method calculates this value based on the ratio between successful and overall attempts. Consequently, the authors tried to answer the ***RQ: What percentage of the recommended coding exercises are relevant?*** by calculating the trade-off between recommended tasks that are consumed by learners and recommended tasks that are captured by the system. In order to evaluate the recommendation accuracy, we use the evaluation metrics precision and recall (Gunawardana & Shani, 2009). These commonly used evaluation metrics are defined as follows:

$$Precision = \frac{tp}{tp+fp} \qquad Recall = \frac{tp}{tp+fn}$$

where t_p (true positive) is the number of coding exercises recommended and clicked by the students, f_p (false positive) is the number of coding exercises recommended but not clicked by the students, and f_n (false negative) is the number of coding exercises clicked by the students but not recommended. The evaluation by precision and recall metrics gives us the accuracy of the recommendations when we use the implemented Elo-rating method for generating the recommendations. The clicks of students on the recommended exercises are interpreted as relevant recommendations. Implementation of other recommendation methods are considered as a future work so that we can compare the accuracy of different recommendation strategies.

References

Gunawardana, A., & Shani, G. (2009). A survey of accuracy evaluation metrics of recommendation tasks. Journal of Machine Learning Research, 10(Dec), 2935-2962.

Hooshyar, D., Ahmad, R. B., Yousefi, M., Fathi, M., Horng, S.-J., & Lim, H. (2016). Applying an online game-based formative assessment in a flowchart-based intelligent tutoring system for improving problem-solving skills. *Computers & Education, 94*, 18-36.

McKenney, S., & Reeves, T. C. (2013). Systematic review of design-based research progress: Is a little knowledge a dangerous thing? *Educational Researcher, 42*(2), 97-100.

Nunamaker Jr, J. F., Chen, M., & Purdin, T. D. (1990). Systems development in information systems research. *Journal of management information systems, 7*(3), 89-106.
Pelánek, R., Papoušek, J., Řihák, J., Stanislav, V., & Nižnan, J. (2017). Elo-based learner modeling for the adaptive practice of facts. *User Modeling and User-Adapted Interaction, 27*(1), 89-118.
Pieterse, V. (2013). *Automated assessment of programming assignments.* Paper presented at the Proceedings of the 3rd Computer Science Education Research Conference on Computer Science Education Research.
Trætteberg, H., & Aalberg, T. (2006). *JExercise: a specification-based and test-driven exercise support plugin for Eclipse.* Paper presented at the Proceedings of the 2006 OOPSLA workshop on Eclipse technology eXchange.
Vega, Y. L. P., Bolaños, J. C. G., Nieto, G. M. F., & Baldiris, S. M. (2012). *Application of item response theory (IRT) for the generation of adaptive assessments in an introductory course on object-oriented programming.* Paper presented at the Frontiers in Education Conference (FIE), 2012.
Wauters, K., Desmet, P., & Van Den Noortgate, W. (2012). Item difficulty estimation: An auspicious collaboration between data and judgment. *Computers & Education, 58*(4), 1183-1193.