Elo-Rating Method: Towards Adaptive Assessment in E-learning

Katerina Mangaroska
Norwegian University of Science and Technology
Trondheim, Norway
Email: mangaroska@ntnu.no

Boban Vesin
University of South-Eastern Norway
Vestfold, Norway
Email: boban.vesin@usn.no

Michail Giannakos
Norwegian University of Science and Technology
Trondheim, Norway
Email: michailg@ntnu.no

***Abstract*—The success of technology enhanced learning can be increased by tailoring the content and the learning resources for every student; thus, optimizing the learning process. This study proposes a method for evaluating content difficulty and knowledge proficiency of users based on modified Elo-rating algorithm. The calculated ratings are used further in the teaching process as a recommendation of coding exercises that try to match the user's current knowledge. The proposed method was tested with a programming tutoring system in object-oriented programming course. The results showed positive findings regarding the effectiveness of the implemented Elo-rating algorithm in recommending coding exercises, as a proof-of-concept for developing adaptive and automatic assessment of programming assignments.**

Keywords*—*recommender systems, programming tutoring system, coding exercises

I. INTRODUCTION

Assessment has an indispensable role in education. On one side, it helps to identify learners' knowledge proficiency and reinforces learners' ability to track their own progress. On another, poor assessment practices could hinder learners' ability to reflect on their progress and misconceptions, causing a significant impact on learning and instruction. Moreover, standardized assessment methods fail to measure meaningful forms of human competence, because teachers and educational systems rarely accommodate for diversity and variability of the learners. Hence, it is hypothesized that adaptive learning systems will revolutionize the learning process by offering content and resources that match learner's current skills and needs [1].

For a system to be considered adaptive, it means that it needs to be able to estimate the difficulty of the learning content (i.e., problems, questions, tasks) and the learner's skills by incorporating an adaptive technique. The adaptability in education is mainly explored through Intelligent Tutoring Systems (ITS) [2] or in the context of Computerized Adaptive Testing (CAT) [3]. However, in this study the primary goal is not to assess learner's knowledge, but to improve the assessment process through practice. For this to happen, the authors had to implement a method that will estimate the difficulty of the learning content and the learner's knowledge.

The proposed methods is based on a modified Elo-rating algorithm, where the outcome does not have a binary value (solved, not solved), because in programming the outcomes cannot be simplified to a binary output. In programming, every solution can be evaluate it from a different aspect (e.g. efficiency of the solution, number of attempts, solving time, etc.) and as such, imposes additional complexity during the assessment process. Hence, the research presented in this paper discusses a proof-of-concept regarding the effectiveness of an implemented Elo-rating algorithm in a programming tutoring system, that acts as a self-correcting system by matching tasks difficulties with learner's proficiency [4]. The research question that the authors aim to answer is: *How effective is the implemented Elo-rating method in recommending coding exercises to learners in introductory programming course?*

II. BACKGROUND

With the advancements in online and blended learning practices, and with the disadvantages of the standardized assessment methods to measure meaningful forms of human competence, the research community emphasized the need for re-conceptualizing the assessment process [5]. Early examples of various types of adaptivity in assessment systems could be seen through web-based individualized dynamic quizzes, adaptive annotations [6] or adaptive questionnaires in computer-assisted surveys [7].

Different adaptive assessment methods within programming courses are mostly related to IRT [8], extensions of the Elo-rating method [9], [10], or Bayesian networks [11]. IRT basic models are build on the assumption of a constant skill [10]. However, these models are not easy to use due to calibration on large samples. Moreover, frequent updates are computationally difficult to deal with when using IRT methods, since new skill estimates have to be calculated for each student after a single response for a single task [10].

A promising alternative to IRT models is the Elo-rating method [12]. The Elo-rating method was originally developed for rating chess players, but lately it has been used in education to overcome some of the gaps imposed by the IRT models. For example, the Elo-rating method does not make any assumptions and it can model skills that change over time [10]. A systematic overview of different variants of the Elo-rating method and their application in education were presented in [10]. In this study, the author demonstrated that the Elo-rating method is inexpensive and simple to implement, as well as suitable for adaptive practices in educational settings.

Recommender systems (RS) are used in education to help learners perform better by considering their preferences and proficiency, and adapt the learning resources to match their skills, goals, and needs [13]. Many researchers examined different recommender techniques and their applicability in personal learning environments [14], [15]. Some of the used techniques include: content-based filtering, user-based collaborative filtering, item-based collaborative filtering, stereotypes or demographics-based collaborative filtering, case-based reasoning, and attribute-based techniques [16].

However, one of the most important benefits that Adaptive Educational Hypermedia and consequently RSs introduced to the learners, is the exploratory learning [17], a process that encourages self-initiated, goal-oriented, and self-regulated learning activities. This is an important skill that learners need to develop, as learning is becoming more blended and distributed across physical and digital learning spaces.

III. THE IMPLEMENTED ELO-RATING METHOD

Considering that the Elo-rating method was originally developed for rating chess players, its adaptation in the field of education assumes that a learner is the player, and the item is the opponent. Furthermore, the Elo-rating of a learner and the content is represented by a number which increases or decreases depending on successful or failed attempts to solve a coding exercise. The basic logic of the formula consists of a learner gaining points if performing above its expectancy level, and losing points if performing below its expectancy level [18]. For example, if a high-rated learner successfully solves a less difficult exercise, then a few rating points will be added to its rating. However, if a lower-rated learner solves an exercise that is above its rank (i.e., a difficult exercise), the learner will receive more ranking points.

The proposed method estimates the probability that a user is able to solve the coding exercise based on its current rank and the difficulty of the coding exercise [19]. This method, implemented in the programming tutoring system, differs from other implementations of the same algorithm. The proposed method calculates this value based on the ratio between the successful attempts and the overall attempts.

A. Recommendation of coding exercises based on generated ratings

The core of the recommendation process based on the proposed Elo-rating method lies in ranking learners' knowledge and recommending coding exercises that match their current proficiency. The learner has an opportunity to choose between recommended or not recommended coding exercises.

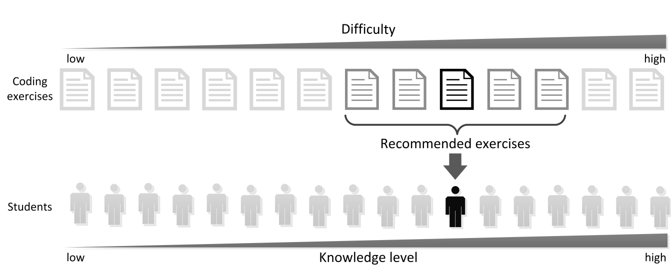

Fig. 1. Recommendation of coding exercises

The learners' ranks and the difficulty level of the coding exercises are re-calculated after every attempt a learner undertakes to solve a coding exercise. Such automated assessment in combination with JUnit testing, could free teachers from checking and correcting hundreds of assignments, because the final grades are calculated automatically and objectively (i.e, avoiding the bias from educator's side).

IV. METHODOLOGY

Setting and participants. For the purpose of this study, first year computer science students at Norwegian University of Science and Technology - NTNU were introduced with a programming tutoring system, that offered interactive learning content for students to learn and practice programming skills. The sample consisted of 67 students who enrolled in an introductory object-oriented programming course in their second semester.

Study design. The students worked on the set of coding exercises known as Programming Course Resource System [20], developed at the University of Toronto. After submitting the coding exercises, the code is being tested against a set of unit tests for a particular problem and the user receives an immediate feedback. To evaluate the recommendation accuracy the authors used the standard precision/recall evaluation metrics [21]. *Precision* is defined as the percentage of recommended items that truly turn out to be relevant (i.e., consumed by the user), while *Recall* is defined as the percentage of relevant (i.e., ground-truth positive) items that have been recommended as positive. In practice, the programming tutoring system creates the ranking of the coding exercises based on the implemented Elo-rating method; hence the top-k coding exercises are recommended to students.

V. RESULTS

A total of 67 students participated in the study, but the final data set contains activities from only 22 most frequent users. The results showed that the probability of the implemented Elo-based algorithm to recommend relevant coding exercises was 0.69 (recall), and the probability that all of the recommended coding exercises were relevant, was 0.70 (precision). To further analyze the behaviour of the students on individual level, *Precision* and *Recall* were calculated for each student individually. The results for the most active students are presented in Figure 2.

Looking at the individual choices of students, one can observe that students could be roughly divided into two groups. The first group includes students that almost blindly followed the recommendations, choosing to solve the exercises that match their proficiency level. The other group of students, almost completely ignored the recommendations, and preferred to select coding exercises based on a topic or concepts they were struggling during learning.

The authors also observed students' intentions in selecting coding exercises. For that purpose, the system have created logs of individual student choices and tracked the changes from the students and the content based on the Elo-rating algorithm. By analysing the systems' log, the authors noticed that in two out of three cases, students tried to solve the coding exercises that closely match their estimated knowledge level.

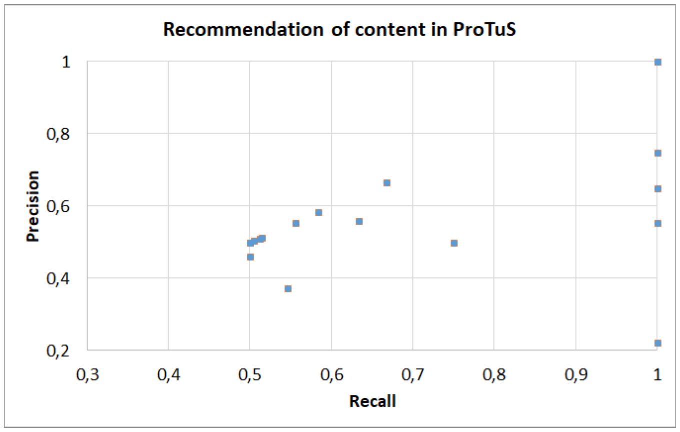

Fig. 2. Precision/recall plot of the performed recommendations

On the other hand, 22% of the students have reached to solve coding exercises that are significantly more challenging than their current level of proficiency, showing ambition to achieve better results.

VI. Discussion and conclusion

The aim of the study is twofold. First, the authors wanted to check if the implemented Elo-rating method is effective in recommending content (i.e., coding exercises) considering the user's proficiency and the task's difficulty. For this purpose, the authors have calculated the classic metrics of precision and recall to estimate the accuracy of the recommendations. Second, the authors wanted to gain more understanding through clickstream data analytics in learner's behavior towards selection of personalized recommendations for the purpose of designing adaptive assessment.

The precision and the recall values, demonstrated that the Elo-rating algorithm has 69% ability to find all relevant coding exercises in the whole set of coding exercises, and 70% precision in recommending the proportion that is actually relevant. Moreover, looking at the students' choices one can notice that majority of the students solved coding exercises that match their knowledge proficiency. This shows that the Elo-rating method effectively pair the level of task difficulty with the learners' knowledge proficiency, allowing the system to adapt to the student's learning strategy.

Finally, the major implication for practice from the findings of this study is that the Elo-rating method could effectively pair items' difficulty with learners' proficiency, leading to recommending relevant items, and towards developing and scaling adaptive assessment in programming courses. Implementing it in practice, this could help educators to sustain higher levels of motivation, performance, and engagement among their students, which are critical components for successful and life-long learning practices.

Acknowledgment

This work was supported by the Norwegian Agency for International Cooperation and Quality Enhancement in Higher Education under the project Smart learning environments (DIG-P44/2019).

References

[1] P. Brusilovsky, L. Pesin, and M. Zyryanov, "Towards an adaptive hypermedia component for an intelligent learning environment," in *International Conference on Human-Computer Interaction*. Springer, 1993, pp. 348–358.

[2] K. Vanlehn, "The behavior of tutoring systems," *International journal of artificial intelligence in education*, vol. 16, no. 3, pp. 227–265, 2006.

[3] R. J. De Ayala, *The theory and practice of item response theory*. Guilford Publications, 2013.

[4] A. Klašnja-Milićević, B. Vesin, M. Ivanović, and Z. Budimac, "Integration of recommendations into java tutoring system," in *The 4th international conference on information technology ICIT 2009 Jordan*, 2009.

[5] K. DiCerbo, V. Shute, and Y. J. Kim, "The future of assessment in technology rich environments: Psychometric considerations," *Learning, design, and technology: An international compendium of theory, research, practice, and policy*, pp. 1–21, 2017.

[6] P. Brusilovsky and S. Sosnovsky, "Individualized exercises for self-assessment of programming knowledge: An evaluation of quizpack," *Journal on Educational Resources in Computing (JERIC)*, vol. 5, no. 3, p. 6, 2005.

[7] C. M. Kehoe and J. E. Pitkow, "Surveying the territory: Gvu's five www user surveys," Georgia Institute of Technology, Tech. Rep., 1997.

[8] J. Y. Park, S.-H. Joo, F. Cornillie, H. L. van der Maas, and W. Van den Noortgate, "An explanatory item response theory method for alleviating the cold-start problem in adaptive learning environments," *Behavior research methods*, pp. 1–15, 2018.

[9] A. Prisco, R. Penna, S. Botelho, N. Tonin, J. Bez *et al.*, "A multidimensional elo model for matching learning objects," in *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2018, pp. 1–9.

[10] R. Pelánek, "Applications of the elo rating system in adaptive educational systems," *Computers & Education*, vol. 98, pp. 169–179, 2016.

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

[12] A. E. Elo, *The rating of chessplayers, past and present*. Arco Pub., 1978.

[13] P. Dwivedi and K. K. Bharadwaj, "e-learning recommender system for a group of learners based on the unified learner profile approach," *Expert Systems*, vol. 32, no. 2, pp. 264–276, 2015.

[14] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: a survey," *Decision Support Systems*, vol. 74, pp. 12–32, 2015.

[15] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, and R. Koper, "Recommender systems in technology enhanced learning," in *Recommender systems handbook*. Springer, 2011, pp. 387–415.

[16] H. Drachsler, H. G. Hummel, and R. Koper, "Personal recommender systems for learners in lifelong learning networks: the requirements, techniques and model," *International Journal of Learning Technology*, vol. 3, no. 4, pp. 404–423, 2008.

[17] D. H. Jonassen, "Hypertext as instructional design," *Educational technology research and development*, vol. 39, no. 1, pp. 83–92, 1991.

[18] M. J. Brinkhuis and G. Maris, "Dynamic parameter estimation in student monitoring systems," *Measurement and Research Department Reports (Rep. No. 2009-1). Arnhem: Cito*, 2009.

[19] R. Pelánek, J. Papoušek, J. Řihák, V. Stanislav, and J. Nižnan, "Elo-based learner modeling for the adaptive practice of facts," *User Modeling and User-Adapted Interaction*, vol. 27, no. 1, pp. 89–118, 2017.

[20] A. Petersen, "Programming course resource system (pcrs)," 2018. [Online]. Available: https://mcs.utm.utoronto.ca/ pcrs/pcrs/

[21] A. Gunawardana and G. Shani, "A survey of accuracy evaluation metrics of recommendation tasks," *Journal of Machine Learning Research*, vol. 10, no. Dec, pp. 2935–2962, 2009.