Measuring task difficulty for online learning environments where multiple attempts are allowed – the Elo rating algorithm approach

Maciej Pankiewicz
Warsaw University of Life Sciences
maciej_pankiewicz@sggw.pl

ABSTRACT

The aim of this research is to examine the accuracy of the estimations performed with the Elo rating system in an online learning environment where multiple attempts are allowed and feedback is provided after every submission. The acquired estimations are compared to the reference difficulty values calculated by the means of the IRT graded response model. The data originates from the RunCode online learning environment (https://runcodeapp.com) developed for the purpose of learning programming skills. The platform has been made available to 299 first semester computer science students with varying initial programming knowledge. There have been 50055 attempts on 76 tasks recorded. Multiple attempts on tasks were allowed, there was no penalty imposed for extra tries and feedback was provided after every submission. High correlation values – up to 0.927 – have been observed for the estimations performed by the Elo rating algorithm. We argue that the design of the Elo algorithm makes it a good choice as the on-the-fly task difficulty estimation method for online learning environments where multiple attempts are allowed and feedback is provided after submission.

Keywords

Task difficulty, Elo, rating algorithm, gamification.

1. INTRODUCTION

There are several methods developed for the purpose of estimating task difficulty that originate to a great extent from the area of item response theory. Some aspects make application of these methods in the context of online learning environments difficult, e.g. computational demands or difficult implementation. Therefore alternative methods of the difficulty estimation are analyzed [1], with the focus on lower computational demands and easier implementation. The Elo rating algorithm is an example of such alternative methods that satisfies these requirements, however, often with the cost of lower (reasonably) estimation accuracy. In online learning environments with formative assessment approach – contrary to knowledge assessment systems – lower accuracy of the estimations may be often accepted. Such learning environments may benefit from the implementation of faster methods, even if the requirement of high estimation accuracy is not met. The Elo rating system has already found several implementations in the educational context [1, 2, 3, 4]. However, most of the up-to-date research focuses on its applications within knowledge testing environments (summative assessment) or online learning platforms (formative assessment) where one attempt is allowed and with examples of task types presenting low complexity, e.g. multi-choice, where it is easy to satisfy the requirement of automated evaluation. The programming assignment is an example of a task type with much higher complexity – it is highly improbable to “guess” the correct answer for such a task type. It is, however, a task type that also satisfies a requirement of an automated evaluation and there are multiple types of automated tests that may be executed on the programming code in order to verify its correctness [5]. It is intuitively expected that on average, the number of attempts needed to correctly solve a programming task is much higher than of e.g. multi-choice task type. But how does it impact the quality of task difficulty estimation? What is the impact of multiple attempts, especially if there is a significant number of tasks available in the system on which learners fail multiple times? Especially online learning environments may benefit from the answer to this question. Dynamically changing number of system users and (or) of collaboratively added tasks make the on-the-fly requirement of the task difficulty estimation hard to satisfy already for a small number of system users and tasks – if using the well-known difficulty estimation methods originating from the area of e.g. item response theory. On the other hand, usage of alternative methods for difficulty estimation may satisfy the on-the-fly requirement, but often with the cost of lower accuracy. This cost however may be often accepted and this research contributes to the question of the above-mentioned compromise between accuracy and on-the-fly calibration requirements in online learning environments.

Maciej Pankiewicz "Measuring task difficulty for online learning environments where multiple attempts are allowed -- the Elo rating algorithm approach" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 648 - 652

2. ESTIMATING DIFICULTY

Models created for the purpose of estimating task difficulty originate mainly from the area of Computerized Adaptive Testing (CAT) domain. These models are used in order to optimize the process of knowledge assessment by lowering the number of tasks and time needed to determine learner's current knowledge level. There are two estimations evaluated: of a task difficulty and of a learner ability. Foundations for the development in this area have been laid by G. Rasch that formulated the single parameter logistic model with difficulty parameter [6, 7, 8]. The model and its variations under the name of the Item Response Theory (IRT) have been since utilized not only in educational [9], but also medical [10] or marketing [11] applications. In the era of the internet education, methods for estimating task difficulty have been used not only for the purpose of the knowledge assessment [12], but are increasingly present in the field of Intelligent Tutoring Systems [13] where they are used for the purpose of matching item difficulty to learner ability in order to optimize the process of knowledge acquisition and achieve so called adaptivity. There are several examples of adaptive online learning systems e.g. for learning factual knowledge in the field of geography [1] or mathematics [2]. There are various methods other than item response theory models used for the purpose of task difficulty estimation that have been evaluated within educational research community e.g. Elo rating algorithm [1, 2, 12], proportion correct [12, 14], learner feedback [12, 15] or expert rating [12]. It has been found that the accuracy of these methods may achieve values described as “accurate-enough” for the purpose of online learning environments [1]. Although in terms of requirements for knowledge assessment systems such accuracy may not always be accepted, it may be a reasonable choice for usage within online learning environments. Above mentioned methods present several weaknesses in terms of their usage within online environments, e.g. require large calibration samples or high computational demands (IRT models), require large number of votes (learner feedback) or availability of experts (expert rating), in the context of their usage within online learning environments some of them may be more reasonable than other, depending on specific aspects of an individual system requirements. This article focuses on the usage of the Elo rating system as it is the algorithmic approach and therefore easier to automate than methods that require involvement of a human, e.g. expert rating or learner feedback. Additionally, it has already been validated as a suitable tool for online learning environments [1]. It has been implemented e.g. in the system with multi-choice questions where one attempt is allowed. This research extends the up-to-date research by presenting results of the analysis performed on the example of the online learning environment with the assignment of higher difficulty level (programming assignment), where multiple attempts are allowed, feedback is provided after every submission and no penalty is imposed for extra tries.

3. ELO RATING ALGORITHM

The Elo rating system [16] has been developed for the purpose of measuring strength of players in chess tournaments. The aim of the algorithm is to calculate players’ rating change after every game. That change depends on outcomes of tournament games. Every player is assigned a rating that is usually a number between 1000 and 3000 that is a subject to change after every game. New rating is calculated by a formula:

$$R_n = R + K(O - P)$$

Where: R_n is the new value of the rating, R – the actual rating, O – game outcome (1 – win, 0 – loss), P – probability of winning the game and constant K – the value for chess tournaments is often 32. The probability of winning P is given as:

$$P = \frac{1}{1 + 10^{\frac{R_o - R_p}{400}}}$$

Where R_p is the rating of a player and R_o is the rating of the opponent. In the context of an online learning environment, we consider a tournament game to be a single submission of a solution, a player – a learner that submits the solution and opponent – a task.

There are three possible outcomes of the chess game (win, loose, draw), but in the context of learning environment we only consider two outcomes: learner wins if the submission receives the maximum score or learner loses if the submission does not receive maximum score.

4. METHODOLOGY

4.1 Programming course

The RunCode online learning environment is an online application that supports automated validation of the correctness of programming code available at https://runcodeapp.com. It provides access to various courses consisting of programming assignments that are grouped into modules for the purpose of clarity. There are several gamification enhancements introduced to the platform aimed at keeping the user engaged.

4.2 Programming assignment

Students learn to code using the RunCode online learning environment by solving programming assignments. Every assignment requires a student to create a code containing a function that will be executed by the test runner in order to check its correctness. Task description defines requirements that the function should meet. Students submit the code containing the function and immediately (after its execution by the test runner) receive score and feedback. Score is calculated as the percentage of the tests, that ended with success to the overall number of tests performed on the code and is presented as a value in the range [0%-100%]. The feedback information is based on the information returned by the test runner and contains errors and warnings (if any) returned by the compiler and results of tests executed by the test runner containing information about the correctness of the submitted code. Only submissions with no errors, no warnings and satisfying requirements of all tests defined by the lecturer receive the maximum score (100%). Multiple submissions are allowed and feedback is provided after every submission.

4.3 Data

The data originates from the gamified course available on the RunCode online learning environment: a platform developed for the purpose of learning programming skills. The RunCode system supports automated evaluation of the submitted programming code. The RunCode platform has been made available as an additional, optional tool during the first-semester *Introduction to programming* course at the Faculty of Applied Informatics and Mathematics, Warsaw University of Life Sciences. The course is mandatory for the students of computer science and is realized in a traditional way – with lectures and computer classes. The main online tool for managing the course resources is the university’s moodle website. Although the RunCode platform usage was not mandatory and results obtained were not included in the final grade, majority of the students decided to use the system on regular basis. The course containing 76 programming tasks has been made available on the RunCode application. The data has been collected during two winter semesters: 2017/2018 and 2018/2019. During this period, 299 students with varying initial programming knowledge used the system. There have been 50055 attempts recorded in total. Multiple attempts were allowed with no penalty imposed on extra tries and feedback was provided immediately after every submission. Students self-elected the order of solving tasks.

5. RESULTS

The data has been collected during two academic years: 2017/2018 and 2018/2019 and contains system usage data that originate from the RunCode platform and results of the survey on the declared initial level of programming knowledge. Before the course started, students took a survey and answered the question about perceived programming skill level – self-evaluation of their knowledge of basic programming concepts. It has been a surprising observation, that about one third of students of the first semester at the Faculty of Informatics declared having completely no previous experience with programming languages and more than a half declared having no (skill level 1) or little (skill level 2) previous programming experience (Table 1).

Table 1. Results of the pre-course survey on programming skill level: 1 – no previous programming experience, 5 – very extensive programming experience.

	1	2	3	4	5
2017 (n=110)	32.7%	24.5%	15.5%	15.5%	11.8%
2018 (n=159)	32.7%	25.8%	22.0%	11.3%	8.2%

The overall engagement of the students, measured as the number of user submissions on the RunCode platform has been presented in Figure 1. The overall engagement of students is considerably high with the average of 178 submissions (attempts) performed by a user.

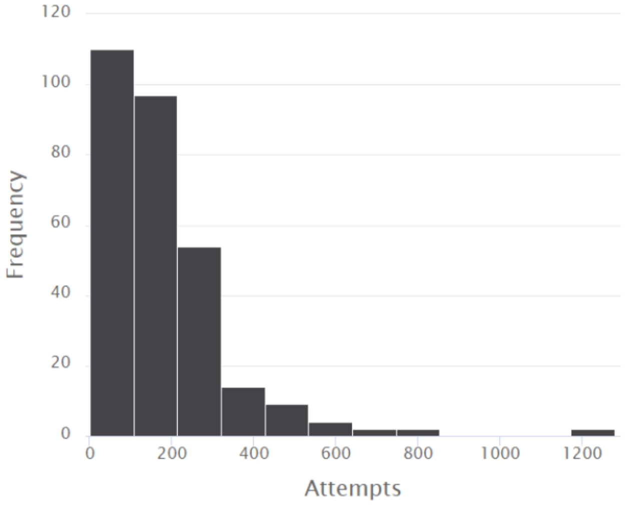

Figure 1. Histogram of the number of attempts performed by users of the RunCode platform.

The following detailed analysis of the submission data (Table 2) for the purpose of clarity has been limited to seven first attempts (ca. 75% of all samples). This limitation is reasonable, as the effects visible on the first seven attempts are in general also reflected in the remaining data (e.g. dropout) with the long tail of even more than 50 attempts on a task. Two important observations may be made basing on the overall view of the data presented in Table 2. Firstly, the number of successful attempts overall is low. On average, the first attempt is correct only in 39% of submissions. If the first attempt was not successful, the success rate for the second submission is 30% and with following attempts, the success rate decreases.

Table 2. The number of correct and incorrect attempts on assignments. The Total column is the cumulative sum of attempts. The Dropout column is the percentage of students that resigned to take another attempt.

Attempts	*Incorrect*	*Correct*	*Total*	*Dropout*
1	8259	5269	13528	-
2	5623	2389	21540	0,030
3	4045	1244	26829	0,059
4	2950	842	30621	0,063
5	2259	493	33373	0,067
6	1766	342	35481	0,067
7	1396	237	37114	0,075

It denotes, that the average difficulty level of tasks available on the platform may be perceived as high. Secondly, despite the fact that users fail to upload successful solution on the first attempt, they feel motivated and do not give up. The dropout rate is very low. Only 3% of the system users give up if the first attempt was not successful. As the number of submission increases, the dropout rate increases but even at the 7th attempt is reasonably low (7%). In order to compare the difficulty estimations calculated by the Elo rating algorithm with the reference values the Pearson's correlation coefficient has been used. Reference values for the following analysis have been calculated by the means of the IRT graded response model [17]. The graded response model is suitable for modelling polytomous response data and has been already introduced e.g. for the purpose of knowledge assessment on open-ended tasks with multiple attempts allowed [18]. It has been found that the estimations of the IRT graded response model are accurate already for sample size of n = 200 [20]. The encoding procedure of polytomous data for the purpose of this analysis was following: the user-task matrix for the *i-th* attempt on task *n* by user *m* has been filled with value of *i*, if the first submission was successful. If the second attempt was successful, the value inserted was *i-1*. Every following attempt needed to achieve the maximum score lowered the inserted value by *1*. In this scenario if a learner does not succeed in a maximum allowed number of attempts, the inserted value is 0. The procedure does not distinguish between not taking the task and exhausting all available attempts with no success. The study on the effects of missing data on the accuracy of estimations performed by the graded response model may be found e.g. in [19]. The reference (IRT) values for the following analysis have been calculated on the full data set. The optimal value of the Elo uncertainty parameter K has been evaluated experimentally, similarly to [12, 14]. The highest correlation with estimation values calculated with the graded response model has been achieved for the value of K = 3. The PlayerRatings R package [21] with default values of the initial rating and rating deviation has been used to perform Elo algorithm calculations and RapidMiner – for the ETL data processing [22]. The highest correlation value – 0.927 – has been observed for cumulative data from three attempts on tasks (Table 3). The correlation calculated only on the data from the first attempt achieves low correlation of the value 0.565.

Table 3. Correlation of the difficulty estimations calculated by Elo algorithm compared to the reference values.

Att.	*1*	*2*	*3*	*4*	*5*	*6*	*7*
Cor.	0,565	0,887	0,927	0,908	0,892	0,873	0,852

With increasing number of attempts, the correlation decreases – correlation value calculated for the cumulative data from 7 attempts is 0.852 (Figure 2).

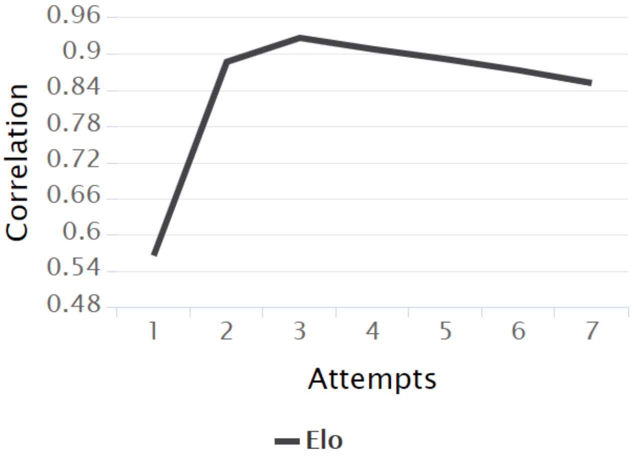

Figure 2. Correlation of the difficulty estimations calculated by Elo algorithm with the reference values.

6. SUMMARY AND DISCUSSION

The aim of this research has been to evaluate the accuracy of the Elo rating algorithm in terms of the task difficulty estimation. The analysis has been performed in order to verify the appropriateness of the method for its usage within online learning environments where multiple attempts are allowed and feedback is provided after every attempt. The source of the data has been the programming course available at https://runcodeapp.com – an online application developed for the purpose of learning programming skills. The analysis has been performed on the sample of 50055 attempts on 76 tasks submitted by 299 learners – first semester students of computer science. The data has been gathered during two academic years: 2017/2018 and 2018/2019. Although usage of the platform was not mandatory, a high level of engagement has been observed – the dropout rate for consecutive attempts was in the range of 3-7%. Students presented varying levels of initial programming knowledge, with about one third declaring no previous experience with programming. The highest correlation of 0.927 has been calculated for data containing three attempts on task. With an increasing number of attempts, the correlation value has slowly decreased. The obtained correlation level may satisfy the requirements of the online learning environment and estimations may be perceived as sufficient. Similar values of correlation have been already obtained in previous research – does it mean that the Elo rating algorithm may be a reasonable choice for estimating difficulty within online learning environments? Under circumstances described in the following, it may be. Contrary to online assessment applications where large calibration samples are required, requirements of online learning environments in terms of the accuracy may not be that strict – although delivering lower accuracy, the Elo algorithm is quick and it is the main advantage. Novel aspects of this analysis concern following factors: 1) it is based on the data originating from the real online learning environment created for the purpose of fostering basic programming skills; 2) allowance of the multiple (unlimited) attempts on task and feedback provided after every submission; 3) high level of the task difficulty observed as the large average number of attempts required to complete the task. There are several considerations that may limit the interpretability of the results and their generalization that may be divided into three elements referring to the RunCode platform, users and task characteristics. The RunCode online learning environment is a gamified internet application. The aim of the implemented gamification elements is to engage platform users and motivate them towards reaching the maximum score on every task. Overall engagement of system users may be described as very high and the gamification may be an important source of the large user contribution. The number of students that give up after an unsuccessful attempt is very low and varies between 3% and 7% for the first 7 attempts (Table 2). It should be a subject for further analysis, if the results may be repeated if the number of dropouts in the data increases. The platform has been made available to the first semester students of computer science enrolled in the *Introduction to programming* course. The variety of the skill level is broad in the analyzed group. Although the first impression may be that students enrolled in the computer science track already have experience with basics of programming, student responses in the survey completed at the beginning of the course do not confirm this suspicion. One-third of the students declares to have absolutely no previous experience with any programming language, but there are also several students that have already mastered the basic programming concepts before joining the course. It is to be analyzed, if the observations from this study are repeated if users present equal (e.g. very low) initial knowledge on the subject. It is also to be considered, that the motivation of the computer science student to succeed in the *Introduction to programming* course may be reasonably higher than of an average user that joins any programming course at any publicly available online learning platform. Although usage of the platform was not mandatory and results obtained were not impacting the final grade, students used the platform very extensively. Therefore, it is to be analyzed if the observations made within a group that focused on the success in the course apply in other contexts. The overall difficulty level of the programming assignment available on the RunCode platform may be described as high. The submission process is very complex in comparison to e.g. multi-choice questions. Even easiest tasks (as perceived by the lecturer) received on average a higher number of attempts than initially expected. It may result from the fact that unexperienced learners that joined the course struggled from the beginning with too many new concepts: not only related to the basic rules of code preparation, but also e.g. to the technical aspects of creating code with usage of the integrated development environment (IDE). There is an additional outcome of the large average number of submissions on a task. The difficulty level may be estimated with higher granularity, even if the number of system users is low. On the other hand, if these tasks were made available outside of the university's course, on an online platform to the public, a high average difficulty level would possibly lead to learners' frustration and it would be expected that the dropout rate will be much higher. Future work will be aimed at comparing other methods of difficulty estimation satisfying the requirements of the on-the-fly calibration.

7. REFERENCES

[1] 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. https://doi.org/10.1007/s11257-016-9185-7

[2] Klinkenberg, S., Straatemeier, M., & van der Maas, H. L. (2011). Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation. Computers & Education, 57(2), 1813-1824. https://doi.org/10.1016/j.compedu.2011.02.003

[3] Papousek, J., Pelanek, R., & Stanislav, V. (2014). Adaptive practice of facts in domains with varied prior knowledge. In Proc. of educational data mining (pp. 6-13).

[4] Pankiewicz, M. & Bator, M. (2019). Elo Rating Algorithm for the Purpose of Measuring Task Difficulty in Online Learning Environments. e-mentor, 5(82), 43-51.

[5] Ala-Mutka, K. M. (2005). A survey of automated assessment approaches for programming assignments. *Computer science education*, *15*(2), 83-102.

[6] Rasch G., 1960, Probabilistic Models for some Intelligence and Attainment Tests, Danish Institute for Education Research, Copenhagen.

[7] Rasch G., 1966, An individualistic approach to item analysis, [w:] P.F. Lazarsfeld, N.W. Henry (eds.) Readings in mathematical social sciences, Cambridge: MIT Press, 89-107.

[8] Rasch G., 1977, On specific objectivity: An attempt at formalising the request for generality and validity of scientific statements, Danish Yearbook of Philosophy, vol. 14, s. 58–94.

[9] Scheerens J., 2003, Educational Evaluation, Assessment, and Monitoring: A Systemic Approach, Swets & Zeitlinger, Lisse–Exton.

[10] Christensen K.B., Kreiner S., Mesbah M., 2013, Rasch Models in Health, ISTE–Wiley, London–Hoboken.

[11] Bechtel G.G., 1985, Generalizing the Rasch model for consumer rating scales, Marketing Science, vol. 4, no. 1, s. 62–73.

[12] 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. https://doi.org/10.1016/j.compedu.2011.11.020

[13] Wauters, K., Desmet, P., & Van Den Noortgate, W. (2010). Adaptive item-based learning environments based on the item response theory: possibilities and challenges. Journal of Computer Assisted Learning, 26(6), 549-562. https://doi.org/10.1111/j.1365-2729.2010.00368.x

[14] Antal, M. (2013). On the use of Elo rating for adaptive assessment. Studia Universitatis Babes-Bolyai, Informatica, 58(1), 29-41.

[15] Chen, C.M., Lee, H.M., & Chen, Y.H. (2005). Personalized e-Learning System Using Item Response Theory, Computers & Education, 44(3), 237-255. https://doi.org/10.1016/j.compedu.2004.01.006

[16] Elo, A. E. (1978). The rating of chess players past and present, New York: Arco Publishing.

[17] Samejima, F. (1969). Estimation of latent ability using a response pattern of graded scores. Psychometrika Monograph, No. 17.

[18] Attali, Y. (2011). Immediate feedback and opportunity to revise answers: Application of a graded response IRT model. Applied Psychological Measurement, 35, 472-479.

[19] Bergner, Y., Choi, I., & Castellano, K. E. (2019). Item Response Models for Multiple Attempts With Incomplete Data. Journal of Educational Measurement, 56(2), 415-436.

[20] Forero, C. G., & Maydeu-Olivares, A. (2009). Estimation of IRT Graded Response Models: Limited Versus Full Information Methods. Psychological Methods, 14(3), 275-299.

[21] Stephenson, A., and Sonas, J. (2019). R package "PlayerRatings". Retrieved from https://CRAN.R-project.org/package=PlayerRatings

[22] Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., & Euler, T. (2006, August). Yale: Rapid prototyping for complex data mining tasks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 935-940).