Adapting Collaboratively by Ranking Solution Difficulty: an Appraisal of the Teacher-Learner Dynamics in an Exploratory Environment

Rômulo C. Silva[1,2], Alexandre I. Direne[2], Diego Marczal[3], Paulo R. B. Guimarães[2], Ângelo S. Cabral[2], and Bruno F. Camargo[2]

[1] Western University of Paraná (UNIOESTE)
[2] Federal University of Paraná
[3] Federal Technological University of Paraná
{romulocesarsilva,dmarczal,guimaraes.prb,angeloscabral}@gmail.com
alexd@inf.ufpr.br
brunofilla_camargo@hotmail.com

Abstract. The work approaches theoretical and implementation issues of a framework aimed at supporting human knowledge acquisition of mathematical concepts. We argue that the problem solving tasks to be carried out by a learner should be ordered according to the matching of two parameters: (1) human skill level and (2) solution difficulty. Both are formally defined here as algebraic expressions based on fundamental principles derived from extensive consultations with experts in pedagogy and cognition. Our general definition of skill level is a rating-based measure that resembles the ones of game mastery scales. Likewise, the solution difficulty includes valuations based on a calibration method that computes mistakes and successes of learners' attempts to deal with the problem. The framework is instantiated by implemented software tools for the domain of logarithmic properties. Finally, we draw conclusions about the suitability of the claims based on a four-highschool-class experiment.

Keywords: rating, exercises calibration, Intelligent Tutoring Systems

1 Introduction

The student's expertise is usually developed by solving exercises that require a set of assessed skills. This is done in both conventional education schools and when applying advanced learning technologies, such as Intelligent Tutoring Systems (ITS). Normally, human teachers detect students' misconceptions when marking tests and exercises. Depending on how much the answer of a question departs from its correct version, two students that missed the same question could be scored different grades for that specific question.

Another aspect that can be used to compose the score is how difficult the question is. The difficulty degree of a question can be measured by the number of students that have skipped or made a mistake in that question. Thus, a student

who finds the correct answer of a question that many missed, probably has more skills than others and the score should reflect that. Conversely, a student who makes a mistake in a question that many were successful to answer, might possess fewer skills. Therefore, when posing questions to a student, it's desirable that an ITS calibrates the difficulties of such questions properly in order to match them against the expertise level of the student.

The student models have become a key element in ITS, supporting the development of individual help and detecting off-task behaviour [1]. The more recent approaches of student displacement behaviour from what is expected are influenced by the other students' behaviour. In this sense, a larger sampling of learners should provide better automatic assessments of a specific learner.

In the construction of student models, an important issue is weather just one or multiple skills will be considered. Some of the proposed models are based on the IRT (Item Response Theory), which is a classical model in psicometrics that assumes that success in every item of a test is determinated by one ability, named θ, referred to as latent trait.

Another desirable aspect in ITS is predicting or prospecting if a learner will be able to answer a question correctly or not before it is actually showed to him or her. This feature allows the exercises to be presented according to the student's skills or rating.

2 Literature Review

Champaign and Cohen propose an algorithm [3] for content sequencing that selects the appropriate learning object to present to a student, based on previous learning experiences of like-minded users. The granularity of sequencing is on the LO level, not exercises or issues. A limitation of the work is that the algorithm was validated only by using simulated students.

Ravi and Sosnovsky [14] propose a calibration method for solution difficulty in ITS based on applying data mining techniques to a student's interaction log. Using the classical bayesian Knowledge Tracing (KT) method [5], the probability that a student has acquired a skill is calculated on the basis of a tentative sequence of exercises for which the soluctions involve a given concept. The logged events are grouped by exercises and classified according to the student's skills. All the data generated by the process is then used to match the sigmoid curve of IRT to connect different students using the standard clustering algorithm k-means.

Schatten and Schmidt-Thieme [15] present the Vygotski Policy Sequencer (VPS), based on the concept of Zone of Proximal Development devised by Vygotski. In this approach, the matrix factorization, which is a method for predicting user rating, is combined with a sequencing policy. This is done in order to select at each time step the content according to the predicted score.

Clement *et al.* [4] propose two algorithms for the tutoring model of ITS. The first, named RiARiT (Right Activity at Right Time), is based on multi-arm bandit techniques [2] such that each activity involves different skills, referred to as

Knowledge Components (KCs). The student model is a generalization of the one used in the bayesian KT method, representing the student's competence level (c_i) by a Real number in the range [0..1]. Furthermore, a reward representing the learning progress is defined by the difference between required KC and c_i. The second algorithm, ZPDES (Zone of Proximal Development and Empirical Success) [4] is a modified version of RiARiT where the calculation of the reward is changed in order to remove the dependence of the student's estimated competence level. The reward becomes a measure of how the success rate is increasing, providing a more predictive choice of activities.

Guzmán and Conejo [10] propose a cognitive assessment model based on IRT for ITS that calibrates the items of a topic (or concept). The method of item calibration is based on the kernel smoothing statistical technique that requires a reduced number of prior students sessions compared to conventional methods. In their approach, each possible answer has a characteristic curve that expresses the probability that a student with a certain knowledge level will more than likely select this answer.

There are several works about rating prediction techniques. Desmarais *et al.* [7] presented a comparative study between different linear models of student skill based on matrix factorization, IRT model and the k-nearest-neighbours approach. The linear models based on matrix factorization make predictions using a subset of the observed performance data for each student to predict the remaining subset, and measure the prediction accuracy. For other works, see [9], [6] and [16].

3 Automatic Calculation of Rating

Rating systems are frequently used in games to measure the players skills and to rank them. Usually, the rating is a number in a range $[minRank, maxRank]$ such that it is very unlikely that a player falls on the extremes. Inspired by game rating systems and taking the performance of other learners, this study proposes Equation 1 to assess iteratively a student's ability.

The following guidelines were adopted: *(1)* each question is scored a difficulty degree with a value in the range [0..10] and the student is rated in the range [1..10] to express his or her expertise level in the subject matter; *(2)* the easier the question, the greater the likelyhood that students will answer it correctly (in this case, a student's rating should have just a small increase if he or she enters the correct answer and should have a large decrease in the case of failure); *(3)* students that are successful in the first attempt to solve a question are scored a higher increment in their expertise level compared to those who need several attempts; *(4)* skipped questions are considered wrong.

Consider Equation 1.The details of its parameters are as follows:

$$R_J^q = R_J^{q-1} + Ak_1\alpha(10 - \frac{9T_J^q}{T_{med}^q}) - Ek_2\beta \times 10\frac{T_J^q}{T_{med}^q} \quad (1)$$

– R_J^q: student J's rating after answering question q;

- R_J^{q-1}: previous student J's rating. $R_J^0 = 5.5$ (initial rating);
- $A = 1$ and $E = 0$ if the student is successful in answering q, otherwise $A = 0$ and $E = 1$;
- T_J^q: number of unsuccessful attempts of student J to answer question q;
- T_{med}^q: median of wrong attempts on question q during classroom time;
- N_a^q: number of students that were successful in answering question q;
- N_e^q: number of students that were unsuccessful in answering question q;
- $\alpha = \frac{1}{N_a^q}$: weight factor to increase rating;
- $\beta = \frac{1}{N_e^q}$: weight factor to decrease rating;
- k_1 and k_2: multiplier factors of rating increase and decrease, respectively, calculated by $k_1 = 1 - \frac{R_J^{q-1}}{10}$ and $k_2 = \frac{R_J^{q-1}-1}{10}$.

Furthermore, $10 - \frac{9T_J^q}{T_{med}^q}$ and $10\frac{T_J^q}{T_{med}^q}$ represent the score of student J in question q in case the answer is correct and incorrect, respectively. There is no limit to the number of attempts T_J^q a student can make to answer a question. However, if there are more than 10 trials, then 10 is taken as the maximum value for calculation purposes. Factors k_1 and k_2 avoid results of the expression in Equation 1 to reach upper and lower bounds of the range [1..10].

Using only the number of attempts and considering that the student usually tries until he or she gets the correct answer, the difficulty degree of a question q can be defined by Equation 2 and its parameters as follows:

$$D^q = \frac{\sum_{J=0}^{J=n} T_J^q}{N_e^q + N_a^q} \tag{2}$$

- D^q: difficulty degree of the question q after an exercise session;
- T_J^q: number of unsuccessful attempts of student J to answer question q. If the number of attempts is greater than 10 trials, then 10 is taken as T_J^q;
- N_e^q and N_a^q are the same as in Equation 1

4 The ADAPTFARMA environment

The ADAPTFARMA (Adaptive Authoring Tool for Remediation of errors with Mobile Learning) prototype software tool is a modified version of FARMA[12], an authoring shell for building mathematical learning objects. In ADAPTFARMA, a learning object (LO) consists of a sequence of exercises following their introduction. The introduction is the theoretical part of a LO where concepts are defined through text, images, sounds and videos. The ADAPTFARMA implementation was carried out aiming its use on the web, either through personal computers or mobile devices.

To build an introduction and its corresponding exercise statements, ADAPTFARMA offers a WYSIWYG (What you See Is What You Get) interface, similar to those of highly interactive word processors. The teacher defines the number of questions related to each exercise. For each question, the teacher-author must

set a reference solution, which is the correct response to the question. ADAPTFARMA allows arithmetic and algebraic expressions to be entered as the reference solution. Under the learner's functioning mode, the tool deals automatically with the equivalence between the learners response and the reference solution.

A feature of ADAPTFARMA is the capability of backtracking the teacher to the exact context in which the learner made a mistake. This gives the opportunity to the teacher to identify the wrong steps performed by the learner and, thus, deal with the causes of the error accordingly. In addition, ADAPTFARMA allows the teacher to view a learner's complete interaction with the tool in a chronological order, in the form of a timeline. The teacher can make a closer monitoring of problem solution from other classrooms, as long as system permission is given through the collaboration mechanisms.

Likewise, learners can backtrack to the context of any of their right or wrong answers in order to reflect about their own solution steps. Additionally, on the collaborative side, it is possible for the teacher to carry out a review of students' responses and then provide them with non-automatic feedback, which can be done by exchanging remote messages through the system.

5 Algorithm for Exercises Sequencing

An important aspect in ITS is how the exercises should be sequenced after they are calibrated in order to match them to the expertise level of the student. At the beginning, the system doesn't have any information about the student. We propose an algorithm for sequencing exercises to be shown in ascending order of difficulty, combined with a mechanism similar to numerical interpolation:

- a minimal sequence of exercises is defined such that always begins with the easiest exercise and finishes with the most difficult one;
- the intermediate level exercises in the minimal sequence are distributed evenly among the easiest and most difficult exercises such that the number of exercises is $\left\lceil \frac{n}{stepsize} \right\rceil$ where n is the total of exercises and the *stepsize* refers to the number of exercises that may be skipped when the student is successful. The *stepsize* can be set by the LO's author;
- the exercises are presented in the minimal sequence order;
- the number of attempts is limited to the average number of attempts obtained in the calibration phase. When the number of attempts is exceeded, the next exercise presented to the student is of a mid range difficulty considering the last exercise correctly answered and the current one.

For example, consider a LO with 30 exercises in ascending order of difficulty $[e_1, e_2, ..., e_{30}]$ and $stepsize = 4$. The minimal sequence of exercises will be $min\_seq =< e_1, e_5, e_9, e_{13}, e_{17}, e_{21}, e_{25}, e_{29}, e_{30} >$, and the exercises will be presented to the student in that order at first. For example, if the student misses e_9 until the attempts are over, then e_7 (of mid range difficulty between e_5 e e_9) is presented. Unlike the calibration phase, the student cannot skip exercises and if he/she continually misses the correct answer, the presentation becomes sequential.

6 Experiment

In order to evaluate the learning effectiveness of the four sequencing strategies, we carried out an experiment with four different classes of highschool students, aging fifteen to seventeen. The same LO about logarithms was applied to all four classes. It was created with the ADAPTFARMA enviroment to include thirty exercises. For each class, the LO was applied with a different sequencing method to order the exercises as follows:

- class A: random sequencing method (RSM);
- class B: teacher-defined sequencing method (TSM);
- class C: difficulty-biased sequencing method (DSM), where the difficulty degree was calculated by Equation 2 using outcome data from the calibration phase of class A;
- Class D: adaptive sequencing method (ASM), using the algorithm described in the previous section

The same pre- and post-tests were applied to all four classes. Students who did not participate in any step have been excluded from the analysis, resulting 119 participants. For the RSM, TSM and DSM methods, there was no limit to the solution attempts while in ASM, the average of attempts in class A was used. The Shapiro-Wilk test was applied to all samples to check for normality. Because only the DSM data passed the normality test (p-value = 0.0827), the pairwise T Student test was applied to it (p-value = 0.532). For the other three, the choice was the Wilcoxon test in order to evaluate the individual sequencing methods. The p-value of RSM, TSM and ASM were 0.0007, < 0.0001 and 0.0037, respectively. All methods, except for DSM, had a significant increase in scores.

The ANOVA method was applied to the pre-test data that showed normality whereas the Kruskal-Wallis, to the others, both to the post-test and to the average difference between pre- and post-tests. The results indicate that there is no significant difference among the four classes in the pre-test scores (p-value = 0.2539). However, there is significant difference in the post-test scores (p-value = 0.00579) and in the average difference between pre- and post-tests scores (p-value = 0.0307), suggesting that RSM, TSM and ASM led to better student performance than DSM. Besides, student performances among the three (RSM, TSM and ASM) were similar. Surprisingly, RSM led to the best performance while DSM, to the worst. This contradicts quite a large proportion of literature research on pedagogic practice, machine-led [8] or otherwise, for developing problem solving skills. Some reasons might explain such a phenomenon:

- the problem-statement ordering is a relevant issue that should be whatched more carefully to verify the influence of tacit knowledge contained in the textual organization of the statement;
- the lack of significant differences between RSM, TSM and ASM is also supported by evidence based on past research findings [11, 13];

– the DSM may have connected some sort of subject matters that caused an increase in the cognitive load, resulting in problem solutions that diverted from the correct ones;
– although most students have participated in the experiment, only the scores of pre- and post-tests accounted for the final student score in the official school records.

7 Conclusion and Future Work

Usually the student's expertise is developed by solving exercises that require a set of assessed skills, including ITS. We proposed an automatic rating system that can be used as an additional tool to assess students. Depending on the number of attempts and the difficulty degree of a question, students can get different scores for the same question.

Also, we proposed an algorithm, referred as ASM, for sequencing exercises that uses difficulty degree combined with a mechanism similar to numerical interpolation. It composes the ADAPTFARMA environment, a web authoring tool with WYSIWYG interface for creating and executing LOs. Taking advantage of it is very easy to change the strategy for exercises sequencing, we carried out a four-highschool-class experiment to test different sequences strategies: RSM, TSM, DSM and ASM. Only DSM had not a significant increase in the students' scores and the RSM had the best performance, demonstrating that problem-statement ordering is a relevant issue that should be researched more carefully in the near future. The ASM had also better performance compared to DSM.

Future research concentrates in adding new features to FARMA in two ways. Firstly, we are working in a deeper approach to user adaptation that includes more dimensions than just the matching between problem difficulty and student skill. One such new feature will be a function for generating problem statements based on teacher-defined problem statement parameters. Secondly, on the interface side, more interaction modes will be available to improve collaboration tasks for monitoring student performance progress.

References

1. Ryan S.J. D. Baker, Adam B. Goldstein, and Neil T. Heffernan. Detecting the Moment of Learning. *LNCS*, 6094(PART I):25–34, 2010. Springer-Verlag Berlin Heidelberg.
2. Sébastin Bubeck and Nicolò Cesa-Bianchi. Regret analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.
3. John Champaign and Robin Cohen. A Model for Content Sequencing in Intelligent Tutorign Systems Based on the Ecological Approach and Its Validation Through Simulated Students. pages 486–491. Association for the Advancement of Artificial Intelligence (AAAI), 2010.

4. Benjamin Clement, Didier Roy, and Pierre-Yves Oudeyer. Online Optimization of Teaching Sequences with Multi-Armed Bandits. In Pardos Z. Mavrikis M. McLaren B.M. Stamper, J., editor, *Proceedings of the 7th International Conference on Educational Data Mining*, pages 269–272, 2014.
5. 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.
6. Maunendra Sankar Desarkar and Sudeshna Sarkar. Rating prediction using preference relations based matrix factorization. In Kalina Yacef Eelco Herder and Stephan Weibelzahl, editors, *Workshop and Poster Proceedings of the 20th Conference on User Modeling, Adaptation, and Personalization (UMAP2012)*, volume 872. http://http://ceur-ws.org/, Jul 2012.
7. Michel C. Desmarais, Rhouma Naceur, and Behzad Beheshti. Linear models of student skills for static data. In Kalina Yacef Eelco Herder and Stephan Weibelzahl, editors, *Workshop and Poster Proceedings of the 20th Conference on User Modeling, Adaptation, and Personalization (UMAP2012)*, volume 872. http://http://ceur-ws.org/, Jul 2012.
8. Alexandre Direne. Authoring intelligent systems for teaching visual concepts. *International Journal of Artificial Intelligence in Education*, 1(4):3–14, 1990.
9. Lucas Drumond, Nguyen Thai-Nghe, Tomáš Horváth, and Lars Schmidt-Thieme. Factorization techniques for student performance classification and ranking. In Kalina Yacef Eelco Herder and Stephan Weibelzahl, editors, *Workshop and Poster Proceedings of the 20th Conference on User Modeling, Adaptation, and Personalization (UMAP2012)*, volume 872. http://http://ceur-ws.org/, Jul 2012.
10. Eduardo Guzmán and Ricardo Conejo. Towards efficient item calibration in adaptive testing. In Liliana Ardissono, Paul Brna, and Antonija Mitrovic, editors, *User Modeling 2005*, volume 3538 of *Lecture Notes in Computer Science*, pages 402–406. Springer Berlin Heidelberg, 2005.
11. N. Major and H. Reichgelt. COCA: A shell for intelligent tutoring systems. In *Proc. of the International Conference on Intelligent Tutoring Systems (ITS92)*, pages 523–530. Springer, 1992.
12. Diego Marczal and Alexandre Direne. Farma: Uma ferramenta de autoria para objetos de aprendizagem de conceitos matemáticos. In *Anais do Simpósio Brasileiro de Informática na Educação*, volume 23, 2012.
13. Antonija Mitrovic. An intelligent SQL tutor on the web. *International Journal of Artificial Intelligence in Education*, 13(3):173–197, 2003.
14. Gautham Adithya Ravi and Sergey Sosnovsky. Exercise difficulty Calibration Based on Student Log Mining. In F. Mödritscher, V. Luengo, E. Lai-Chong Law, and U. Hoppe, editors, *Proceedings of DAILE'13: Workshop on Data Analysis and Interpretation for Learning Environments*, Villard-de-Lans (France), Janeiro 2013.
15. Carlotta Schatten and Lars Schmidt-Thieme. Adaptive Content Sequencing without Domain Information. *6th International Conference on Computer based Education*, April 2014.
16. Avi Segal, Ziv Katzir, Kobi Gal, Guy Shani, and Bracha Shapira. EduRank: A Collaborative Filtering Approach to Personalization in E-learning. In Pardos Z. Mavrikis M. McLaren B.M. Stamper, J., editor, *Proceedings of the 7th International Conference on Educational Data Mining*, pages 68–75, 2014.