

Matrix Factorization Feasibility for Sequencing and Adaptive Support in ITS

Carlotta Schatten, Ruth Janning, Lars Schmidt-Thieme
Information Systems and Machine Learning Lab
University of Hildesheim, Germany
schatten, janning, schmidt-thieme@ismll.uni-hildesheim.de

Manolis Mavrikis
London Knowledge Lab
University of London, UK
m.mavrikis@lkl.ac.uk

ABSTRACT

Performance prediction has the potential of ameliorate the student model of an Intelligent Tutoring System by predicting whether a student mastered or not a specific set of skills. Recently, it has been shown, by means of a simulated learning process, how performance prediction methods based on Matrix Factorization can be used for continuous score prediction and for sequencing contents through a policy inspired by Vygotsky's concept of the Zone of Proximal Development. In this paper we discuss the feasibility of the approach analysing a commercial system dataset. We evaluate performances of the score predictor and feasibility of the Vygotsky policy for sequencing tasks and providing adaptive support.

Keywords

Matrix Factorization, Sequencing, Adaptive Support

1. INTRODUCTION & BACKGROUND

In Intelligent Tutoring Systems (ITS), adaptive sequencers can take past student performance into account to select the next task which best fits the student's learning needs. Simple sequencing policies rely on assumptions such as that a student will be able to solve an exercise of the achieved difficulty level but not the more difficult ones without having completed ones of the previous level. This can be problematic as it requires students to go through all the topics in the current level even if they can answer them successfully with the first attempt. Although the power-law-of-practice [4] would suggest that students should be provided with several opportunities to practice, unnecessary repetition can be detrimental in that it can lead to student frustration and influence their perception of the reliability of the system. One way to approach the problem is based on assessing the student skills and matching them to the required skills and difficulties of the available tasks. For example, in [2] the less known skills by the students are selected to be practiced in the next session. In this scenario two problems arise: 1. Tagging tasks with required skills necessitates experts and thus is a time-consuming, costly process, and, especially for fine-grained skill levels, also potentially subjective. 2. Learning adaptive sequencing models requires online experiments with students and specific data collection policies, that consists, at the beginning, in many randomly proposed tasks. Problem 1. extends also to common performance prediction methods and their extensions: Bayesian Knowledge Tracing (BKT) [1] and Performance Factors Analysis (PFA)[5].

On the contrary, Matrix Factorization (MF), the algorithm

we use for performance prediction, is domain agnostic. Its most common use is for Recommender Systems and in previous work [6] we showed how a score prediction method and a simple policy, inspired by Vygotsky's concept of Proximal Development, could be used for ameliorating sequencing in a simulated environment. Despite its plausibility, applying this sequencer in a real use case of an already established ITS and real students requires design decisions that are not well documented. In this paper we show promising preliminary results and work in progress toward the use of the sequencer in a multi-topic commercial ITS. Moreover, we discuss how the performance prediction indications could be used to help hint provision, where Machine Learning was also applied [3]. Our main goal is to present first results towards the integration of the sequencer presented in [6] into an open architecture while discussing its feasibility.

2. FEASIBILITY DISCUSSION

In this section we discuss how the MF can be applied to a commercial system which has over 1000 lessons in 20 topics and was adapted to be used in several countries like United Kingdom, USA, and Russia. We performed a practical feasibility study using a dataset that is composed by data collected from children from five to fourteen years using the ITS in classrooms and homes. A lesson is composed of test and exercise sessions. The exercise session consists of approximately 10 exercises on a topic and specific learning objectives. While trying to solve those exercises a student can consult several hints, one of those is the bottom-out hint, which displays the solution. In order to pass the exercise a student must achieve a score of 7 out of 10 (7/10) that allows them to pass to the test session. There students have to show what they learned answering 5 questions with a score greater than 6/10. The lesson sequencing policy relies on the assumption that a student will be able to solve the exercises of the achieved difficulty level but not the more difficult ones without having completed all the lessons of the previous level. In contrast to state-of-the-art performance prediction, where the main task is to predict the student's correct at first attempt answer, the commercial system uses the score as student's performance measurement. The data granularity level is low if compared with benchmark systems, since we possess a single score record for the 10 questions of the exercises and one record for the five test questions.

2.1 Performance Prediction Feasibility

In this paper we use Matrix Factorization (MF) as score predictor since we do not possess Knowledge Component

Table 1: Performance Prediction Error

Experiments, score range [0,1]	RMSE, \pm SD over five experiments
Global average	0.3032796
Biased User-Item Exercise	$0.2639167 \pm 3.6989 \cdot 10^{-5}$
Exercise Preprocessing	$0.26061115 \pm 5.97504 \cdot 10^{-5}$

Table 2: Dataset Statistics

Number of Items (Exercise/Topic)	9091/4169
Number of Students	258391
Total Student-Item Interactions	30813070
Total Exercise sessions	17512972
Exercise passed (Score 70-99)	9520278 i.e. 54
Gaming the system (Score 100 + Bottom-out hint)	3988891 i.e. 23%
Total Test sessions	13300098
Test session passed (Score 60-99)	4378461 i.e. 33%
Average score obtained	8.1

information. The matrix $Y \in \mathbb{R}^{n_s \times n_c}$ can be seen as a table of n_c total tasks and n_s students used to learn the students' model, where for some tasks and students performance measures are given. MF decomposes the matrix Y in two other ones $\Psi \in \mathbb{R}^{n_c \times P}$ and $\Phi \in \mathbb{R}^{n_s \times P}$, so that $Y \approx \hat{Y} = \Psi\Phi^T$. Ψ and Φ are matrices of latent features. Their elements are learned with gradient descend from the given performances. This allows computing the missing elements of Y for each student i in each task j of a dataset D . The optimization function is represented by: $\min_{\psi_j, \varphi_i} \sum_{i,j \in D} (y_{ij} - \hat{y}_{ij})^2 + \lambda(\|\Psi\|^2 + \|\Phi\|^2)$, where one wants to minimize the regularized squared error on the set of known scores. MF prediction is computed as:

$$\hat{y}_{ij} = \mu + \mu_{cj} + \mu_{si} + \sum_{p=0}^P \varphi_{ip} \psi_{jp} \quad (1)$$

where μ , μ_c and μ_s are respectively the average performance of all tasks of all students, the learned average performance of a content, and learned average performance of a student. The two last mentioned parameters are also learned with the gradient descend algorithm. We followed the standard approach in the field to divide the dataset temporally in two thirds for training and one third for testing, evaluating the performances with the Root Mean Square Error (RMSE). The score, as in [6], is represented in a continuous interval which goes from zero to one. In Table 1 we present Global Average, i.e. a worst case predictor that assumes students will always perform equally to the global score average computed on the training dataset. The Biased User-Item predictor, instead, uses only the biases μ , μ_s , and μ_c of Eq. 1, i.e. the latent features number P is set to zero. Consequently, out of Table 1 one can see the contribution of the single components of Eq. 1 in ameliorating the prediction. According to the results, the dataset is suitable for the task and MF is able to predict a continuous interval performance in a multiple-topic scenario.

2.2 Sequencing and Hinting Policy Feasibility

The Vygotsky Policy based Sequencer (VPS) in [6] is composed of two components: the Vygotsky Policy (VP) and a Performance Predictor. Given MF as predictor, we want to sequence tasks in a way that attempts to keep students in the so-called zone of proximal development (ZPD), which, in our context, we associate with tasks that are neither too easy nor too difficult to accomplish without much help. This concept is formalized by the following formula: $c^{t*} = \operatorname{argmin}_c |y_{th} - \hat{y}^t(c)|$, where y_{th} is a threshold score that will challenge the students and keep them in the ZPD. The policy will select at each time step the content c^{t*} with the predicted score \hat{y}^t at time t most similar to y_{th} . Considering our use case with a score range for passing of 6-10/10, y_{th} should be set in the middle of the interval, so

that the most exercise selected are predicted with a score of 8. This avoids that, in case of no available tasks predicted with exactly y_{th} , the policy does not select exercises which are out of the score range for passing and consequently minimizing the risk of MF incorrect prediction. With an RMSE of ± 2.6 (Table 1), the selected lessons are approximately always in the aforementioned range.

Another use of the performance prediction is to enhance feedback provision to students as it provides the possibility of developing 'task-independent' adaptive support, i.e. hints that relate to students' interaction overall rather than the specific problem solving steps. At least in the case of the commercial ITS under investigation, problem solving steps are dealt by different components and in fact operate as individual learning objects. Examples of such feedback include the provision of support at the beginning or end of the exercise but also during an exercise if, for example, there is no task-specific help to provide. Accordingly, when students start their experience, it is helpful to provide suggestions about which topic(s) to study based on the MF prediction. The topic having the most tasks in the ZPD, should be proposed. During an exercise, if students attempt to ask for help but the prediction above indicates that they do not seem to need it, the system can restrict help depending on the current answers and attempts on the exercise as in [3].

3. FUTURE WORK

In this paper we discussed the feasibility of employing a Matrix Factorization prediction to sequencing and providing adaptive support. Our plan is to apply the sequencer as task and hint sequencer. However, a still open issue is how to evaluate the contribution of the VPS. Considering the number of exercises passed and failed reveals only a part of the incorrectly sequenced tasks, i.e. the too difficult/failed ones. We believe there are possible improvements on other aspects of the interaction such as a reduction of the 'gaming the system' behaviour (see Table 2) as indicated by the tasks that are achieved with 100%, having accessed the bottom-out-hint and not spending enough time to reflect on it.

Acknowledgements. This research was funded from the FP7 STREP iTalk2Learn (www.iTalk2Learn.eu).

4. REFERENCES

- [1] A. Corbett and J. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *UMAI*, 1994.
- [2] K. Koedinger, P. Pavlik, J. Stamper, T. Nixon, and S. Ritter. Avoiding problem selection thrashing with conjunctive knowledge tracing. In *EDM*, 2011.
- [3] M. Mavrikis. Data-driven modelling of students' interactions in an ILE. In *EDM*, 2008.
- [4] A. Newell and P. S. Rosenbloom. Mechanisms of skill acquisition and the law of practice. *Cognitive skills and their acquisition*.
- [5] C. H. Pavlik, P. and K. Koedinger. Performance factors analysis-a new alternative to knowledge tracing. In *AIED*, 2009.
- [6] C. Schatten and L. Schmidt-Thieme. Adaptive content sequencing without domain information. In *CSEDU*, 2014.