Factorization Models for Forecasting Student Performance

Nguyen Thai-Nghe, Tomáš Horváth and Lars Schmidt-Thieme, University of Hildesheim, Germany

Predicting student performance (PSP) is one of the educational data mining task, where we would like to know how much knowledge the students have gained and whether they can perform the tasks (or exercises) correctly. Since the student's knowledge improves and cumulates over time, the sequential (temporal) effect is an important information for PSP. Previous works have shown that PSP can be casted as rating prediction task in recommender systems, and therefore, factorization techniques can be applied for this task. To take into account the sequential effect, this work proposes a novel approach which uses tensor factorization for forecasting student performance. With this approach, we can personalize the prediction for each student given the task, thus, it can also be used for recommending the tasks to the students. Experimental results on two large data sets show that incorporating forecasting techniques into the factorization process is a promising approach.

1. INTRODUCTION

Predicting student performance, one of the tasks in educational data mining, has been taken into account recently [Toscher and Jahrer 2010; Yu et al. 2010; Cetintas et al. 2010; Thai-Nghe et al. 2011]. It was selected as a challenge task for the KDD Cup 2010[1] [Koedinger et al. 2010]. Concretely, predicting student performance is the task where we would like to know how the students learn (e.g. generally or narrowly), how quickly or slowly they adapt to new problems or if it is possible to infer the knowledge requirements to solve the problems directly from student performance data [Corbett and Anderson 1995; Feng et al. 2009], and eventually, we would like to know whether the students perform the tasks (exercises) correctly (or with some levels of certainty). As discussed in Cen et al. [2006], an improved model for predicting student performance could save millions of hours of students' time and effort in learning algebra. In that time, students could move to other specific fields of their study or doing other things they enjoy. From educational data mining point of view, an accurate and reliable model in predicting student performance may replace some current standardized tests, and thus, reducing the pressure, time, as well as effort on "teaching and learning for examinations" [Feng et al. 2009; Thai-Nghe et al. 2011].

To address the problem of predicting student performance, many papers have been published but most of them are based on traditional classification/regression techniques [Cen et al. 2006; Feng et al. 2009; Yu et al. 2010; Pardos and Heffernan 2010]. Many other works can be found in Romero et al. [2010]. Recently, [Thai-Nghe et al. 2010; Toscher and Jahrer 2010; Thai-Nghe et al. 2011] have proposed using recommendation techniques, e.g. matrix factorization, for predicting student performance. The authors have shown that *predicting student performance can be considered as rating prediction* since the *student, task*, and *performance* would become *user, item*, and *rating* in recommender systems, respectively. We know that learning and problem-solving are complex cognitive and affective processes that are different to shopping and other e-commerce transactions, however, as discussed in Thai-Nghe et al. [2011], the factorization models in recommender systems are implicitly able to encode latent factors of students and tasks (e.g. "slip" and "guess"), and especially in case where we do not have enough meta data about students and tasks (or even we have not enough background knowledge of the domain), this mapping is a reasonable approach.

[1] http://pslcdatashop.web.cmu.edu/KDDCup/

Author's address: Information Systems and Machine Learning Lab (ISMLL), University of Hildesheim, Marienburger Platz 22, 31141 Hildesheim, Germany. Emails: {nguyen, horvath, schmidt-thieme}@ismll.de

Moreover, from the pedagogical aspect, we expect that students (or generally, learners) can improve their knowledge over time, thus, the temporal/sequential information is an important factor in predicting student performance. Thai-Nghe et al. [2011] proposed using three-mode tensor factorization (on student/task/time) instead of matrix factorization (on student/task) to take the temporal effect into account.

Inspired from the idea in Rendle et al. [2010], which used matrix factorization with Markov chains to model sequential behavior of the user in e-commerce area, and also inspired from the personalized forecasting methods [Thai-Nghe et al. 2011], we propose a novel approach, *tensor factorization forecasting*, to model the sequential effect in predicting student performance. Thus, we bring together the advantages of both forecasting and factorization techniques in this work. The proposed approach can be used not only for predicting student performance but also for recommending the tasks to the students, as well as for the other domains (e.g. recommender systems) in which the sequential effect should be taken into account.

2. RELATED WORK

Many works can be found in [Romero and Ventura 2006; Baker and Yacef 2009; Romero et al. 2010] but most of them relied on traditional classification/regression techniques. Concretely, Cen et al. [2006] proposed a semi-automated method for improving a cognitive model called Learning Factors Analysis that combines a statistical model, human expertise and a combinatorial search; Thai-Nghe et al. [2009] proposed to improve the student performance prediction by dealing with the class imbalance problem, using support vector machines (i.e., the ratio between passing and failing students is usually skewed); Yu et al. [2010] used linear support vector machines together with feature engineering and ensembling techniques for predicting student performance. These methods work well in case we have enough meta data about students and tasks.

In student modeling, Corbett and Anderson [1995] proposed the Knowledge Tracing model, which is widely used in this domain. The model assumes that each skill has four parameters: 1) initial (or prior) knowledge, which is the probability that a particular skill was known by the student before interacting with the tutoring systems; 2) learning rate, which is the probability that student's knowledge changes from unlearned to learned state after each learning opportunity; 3) guess, which is the probability that a student can answer correctly even if he/she does not know the skill associated with the problem; 4) slip, which is the probability that a student makes a mistake (incorrect answer) even if he/she knows the required skills. To apply the knowledge tracing model for predicting student performance, the four parameters need to be estimated either by using Expectation Maximization method [Chang et al. 2006] or by using Brute-Force method [Baker et al. 2008]. Pardos and Heffernan [2010] propose a variant of knowledge tracing by taking individualization into account. These models explicitly take into account the "slip" and "guess" latent factors.

Recently, researchers have proposed using recommender system techniques (e.g. matrix factorization) for predicting student performance [Thai-Nghe et al. 2010; Toscher and Jahrer 2010]. The authors have shown that predicting student performance can be considered as rating prediction since the *student, task*, and *performance* would become *user, item*, and *rating* in recommender systems, respectively; Extended from these works, Thai-Nghe et al. [2011] proposed tensor factorization models to take into account the sequential effect (for modeling how student knowledge changes over time). Thus, the authors have modeled the student performance as a 3-dimensional recommender system problem on *(student, task, time)*.

In this work, the problem setting is similar to our previous work [Thai-Nghe et al. 2011], however, we introduce two new methods - tensor factorization forecasting models - for predicting student performance.

3. PREDICTING STUDENT PERFORMANCE (PSP)

The problem of predicting student performance is to predict the likely performance of a student for some exercises (or part thereof such as for some particular steps) which we call the *tasks*. The task could be to solve a particular step in a *problem*, to solve a whole problem or to solve problems in a *section* or *unit*, etc.

Detailed descriptions can be found in [Thai-Nghe et al. 2011]. Here, we are only interested in three features, e.g. student ID, task ID, and time ID.

More formally, let S be a set of students, I be a set of tasks, and $P \subseteq \mathbb{R}$ be a range of possible performance scores. Let $\mathcal{D}^{train} \subseteq (S \times I \times P)^*$ be a sequence of observed student performances and $\mathcal{D}^{test} \subseteq (S \times I \times P)^*$ be a sequence of unobserved student performances. Furthermore, let

$$\pi_p : S \times I \times P \to P, \quad (s, i, p) \mapsto p$$

and

$$\pi_{s,i} : S \times I \times P \to S \times I, \quad (s, i, p) \mapsto (s, i)$$

be the projections to the performance measure and to the student/task pair. Then the problem of student performance prediction is, given $\mathcal{D}^{train}$ and $\pi_{s,i}(\mathcal{D}^{test})$, to find

$$\hat{p} = \hat{p}_1, \hat{p}_2, \ldots, \hat{p}_{|\mathcal{D}^{test}|}$$

such that

$$err(p, \hat{p}) := \sum_{l=1}^{|\mathcal{D}^{test}|} (p_l - \hat{p}_l)^2$$

is minimal with $p := \pi_p(\mathcal{D}^{test})$. Some other error measures could also be considered.

As discussed in Thai-Nghe et al. [2011], the problem of predicting student performance can be *i) casted as rating prediction task in recommender systems* since s, i and p would be *user, item* and *rating*, respectively, and *ii) casted as forecasting problem* (illustrated in Figure 1b-top) to deal with the potentially sequential effects (e.g. describing how students gain experience over time) which is discussed in this work. An illustration of predicting student performance which takes the data sequence into account is presented in Figure 1a. Figure 1b-bottom is an example of representing student performance data in a three-mode tensor.

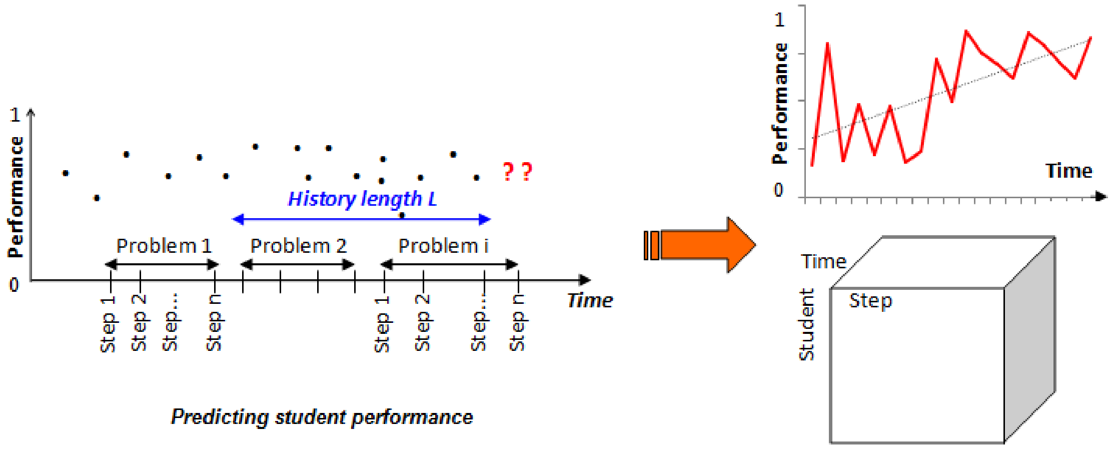

Fig. 1. An illustration of casting predicting student performance as forecasting problem, which uses all historical performance data controlled by the history length L to forecast the next performance

4. TENSOR FACTORIZATION FORECASTING

In this work, we will use three-mode tensor factorization which is a generalization of matrix factorization. Given a three-mode tensor $\mathbf{Z}$ of size $U \times I \times T$, where the first mode describes U students, the second mode describes I tasks (problems), and the third mode describes the time. Then $\mathbf{Z}$ can be written as a sum of rank-1 tensors by using CANDECOM-PARAFAC [Carroll and Chang 1970; Harshman 1970; Kolda and Bader 2009]:

$$\mathbf{Z} \approx \sum_{k=1}^{K} \lambda_k w_k \circ h_k \circ q_k \tag{1}$$

where $\circ$ is the outer product; $\lambda_k \in \mathbb{R}^+$; and each vector $w_k \in \mathbb{R}^U$, $h_k \in \mathbb{R}^I$, and $q_k \in \mathbb{R}^T$ describes the latent factor vectors of the student, task, and time, respectively (see the articles [Kolda and Bader 2009; Dunlavy et al. 2011] for details). In this work, these latent factors are optimized for root mean squared error (RMSE) using stochastic gradient descent [Bottou 2004].

As mentioned in the literature, "the more the learners study the better the performance they get", and the knowledge of the learners cumulates over time, thus the temporal effect is an important factor to predict the student performance. We adopt the ideas in the previous works [Dunlavy et al. 2011][2], [Thai-Nghe et al. 2011; Thai-Nghe et al. 2011] to incorporate forecasting model into the factorization process, which we call tensor factorization forecasting.

For simplification purpose, we apply the moving average approach (the unweighted mean of the previous n data points [Brockwell and Davis 2002]) with a period L on the time mode. The performance of student u given task i is predicted by:

$$\hat{p}_{uiT^*} = \sum_{k=1}^{K} w_{uk} h_{ik} \Phi_{T^*k} \tag{2}$$

where

$$\Phi_{T^*k} = \frac{\sum_{t=T^*-L}^{T^*-1} q_{tk} p_t}{L} \tag{3}$$

where T^* is the current time in the sequence; q_{tk} and p_t are the time latent factor and the student performance of the previous time, respectively; L is the number of steps in the history to be used by the model (refer back to Figure 1 to see the value of L). We call this method **TFMAF** (Tensor Factorization - Moving Average Forecasting).

As shown in [Toscher and Jahrer 2010; Thai-Nghe et al. 2011], the prediction result can be improved if one employs the biased terms into the prediction model. In educational setting, those biased terms are *"student bias"* which models how good a student is (i.e. how likely is the student to perform a task correctly), and *"task bias"* which models how difficult/easy the task is (i.e. how likely is the task to be performed correctly). To take into account the "student bias" and "task bias", the prediction function (2) now becomes:

$$\hat{p}_{uiT^*} = \mu + b_u + b_i + \sum_{k=1}^{K} w_{uk} h_{ik} \Phi_{T^*k} \tag{4}$$

where μ is the global average (average performance of all students and tasks in $\mathcal{D}^{train}$):

$$\mu = \frac{\sum_{p \in \mathcal{D}^{train}} p}{|\mathcal{D}^{train}|} \tag{5}$$

[2] This work used tensor factorization for link prediction

b_u is student bias (average performance of student u deviated from the global average):

$$b_u = \frac{\sum_{p^u \in \mathcal{D}^{train}} (p^u - \mu)}{|p^u \in \mathcal{D}^{train}|} \tag{6}$$

and b_i is task bias (average performance on task i deviated from the global average):

$$b_i = \frac{\sum_{p^i \in \mathcal{D}^{train}} (p^i - \mu)}{|p^i \in \mathcal{D}^{train}|} \tag{7}$$

Moreover, in e-commerce area, Rendle et al. [2010] have used matrix factorization with Markov chains to model sequential behavior by learning a transition graph over items that is used to predict the next action based on the recent actions of a user. The authors proposed using previous "basket of items" to predict the next "basket of items" with high probabilities that the users might want to buy. However, in educational environment, one natural fact is that the performance of the students not only depend on the recent knowledge (e.g. the knowledge in the previous problems or lessons, which act as "previous basket of items") but also depend on the cumulative knowledge in the past that the students have studied. Thus, we need to adapt this method by using all previous performances which are controlled by history length L (see Figure 1) for forecasting the next performance.

The Φ_{T^*k} in equation (3) now becomes:

$$\Phi_{T^*k} = \frac{\sum_{t=T^*-L}^{T^*-1} h'_{tk} q_{tk} p_t}{L} \tag{8}$$

where h'_{tk} is the latent factor of the previous solved task in the sequence. We call this method **TFF** (Tensor Factorization Forecasting).

5. EVALUATION

In this section, we first present two real-world data sets, then we describe the baselines for comparison. We show how we set up the models, and finally, the results of tensor factorization forecasting are discussed.

5.1 Data sets

We use 2 real world data sets which are collected from the Knowledge Discovery and Data Mining Challenge 2010[3]. These data sets, originally labeled "Algebra 2008-2009" and "Bridge to Algebra 2008-2009" will be denoted "Algebra" and "Bridge" for the remainder of this paper. Each data set is split into a train and a test partition as described in Table I. The data represents the log files of interactions between students and the tutoring system. While students solve math related problems in the tutoring system, their activities, success and progress indicators are logged as individual rows in the data sets.

Table I. Original data sets

Data set	#Attributes	#Instances
Algebra-2008-2009 train	23	8,918,054
Algebra-2008-2009 test	23	508,912
Bridge-to-Algebra-2008-2009 train	21	20,012,498
Bridge-to-Algebra-2008-2009 test	21	756,386

The central element of interaction between the students and the tutoring system is the *problem*. Every problem belongs into a hierarchy of *unit* and *section*. Furthermore, a problem consists of many individual

[3] http://pslcdatashop.web.cmu.edu/KDDCup/

steps such as calculating a circle's area, solving a given equation, entering the result and alike. The field *problem view* tracks how many times the student already saw this problem. The other attributes we have not used in this work. Target of the prediction task is the *correct first attempt* (CFA) information which encodes whether the student successfully completed the given step on the first attempt (CFA = 1 indicates correct, and CFA = 0 indicates incorrect). The prediction would then encode the certainty that the student will succeed on the first try.

As presented in Thai-Nghe et al. [2010], these data sets can be mapped to $user, item$, and *rating* in recommender systems. The student becomes the *user*, and the correct first attempt (CFA) becomes the *rating*, bounded between 0 and 1. The authors also presented several options that can be mapped to the *item*. In this work, the *item* refers to a *solving-step*, which is a combination (concatenation) of *problem hierarchy* (PH), *problem name* (PN), *step name* (SN), and *problem view* (PV). The information of $student, task$, and $performance$ is summarized in Table II.

Table II. Information of students, tasks (solving-steps), and performances (CFAs)

Data set	**#Student (as User)**	**#Task (as Item)**	**#Performance (as Rating)**
Algebra	3,310	1,416,473	8,918,054
Bridge	6,043	887,740	20,012,498

5.2 Evaluation metric and model setting

Evaluation metric: The root mean squared error (RMSE) is used to evaluate the models.

$$RMSE = \sqrt{\frac{\sum_{ui \in \mathcal{D}^{test}} (p_{ui} - \hat{p}_{ui})^2}{|\mathcal{D}^{test}|}} \tag{9}$$

Baselines: We use the *global average* as a baseline, i.e. predicting the average of the target variable from the training set. The proposed methods are compared with other methods such as *student average* (user average in recommender systems), *biased-student-task* (this method originally is user-item-baseline in Koren [2010]). Moreover, we also compare the proposed approach with *matrix factorization* (MF) since previous works [Toscher and Jahrer 2010; Thai-Nghe et al. 2010] have shown that MF can produce promising results. For MF, the mapping of *user* and *item* as the following:

student $\mapsto$ *user*;
Problem hierarchy (unit, section), problem name, step name, problem view $\mapsto$ *item*;
performance $\mapsto$ *rating*

Hyper parameter setting: Hyper parameter search was applied to determine the hyper parameters[4] for all methods (e.g, optimizing the RMSE on a holdout set). We will report later the hyper parameters for some typical methods (in Table IV). Please note that we have not performed the significance test (t-test) because the real target variables of the two data sets from KDD Challenge 2010, until now, have not been published yet. We have to submit the results to the KDD Challenge 2010 website to get the RMSE score. Thus, all the results reported in this study are the RMSE score from this website (it is still opened for submission after the challenge). Of course, one can use the internal split (e.g. splitting the training set to sub-train and sub-test) but we have not experimented in this way since we would like to see how good the results of our approach are compared to the other approaches on the given data sets.

[4] Using similar approach described in [Thai-Nghe et al. 2010]

Dealing with cold-start problem: To deal with the "new user" (new student) or "new item" (new task), e.g., those that are in the test set but not in the train set, we simply provide the global average score for these new users or new items. However, using more sophisticated methods, e.g. in [Gantner et al. 2010], can improve the prediction results. Moreover, in the educational environment, the cold-start problem is not as harmful as in the e-commerce environment where the new users and new items appear every day or even hour, thus, the models need not to be re-trained continuously.

5.3 Results

To justify why forecasting method can be a choice in predicting student performance (especially embedding in the factorization process) and how the sequential (temporal) information affects to the performance of the learners, we plot the student performance on the $y-axis$ and the problem ID (in sequence) on the $x-axis$. However, in the experimental datasets, the true target variable (the actual performance) for each single step is encoded by binary values, i.e., 0 (incorrect) and 1 (correct), thus, the student performance does not show the trend line when we visualize these data sets.

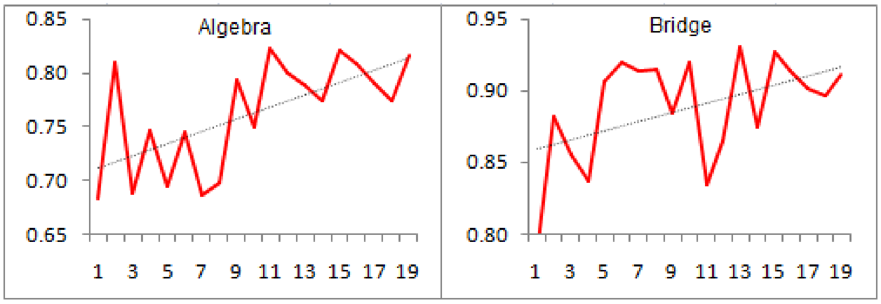

Fig. 2. Sequential effect on the student performance: $y-axis$ is the average of correct performances and $x-axis$ is the sequence of problems (ID) aggregated from the steps. Typical results of Unit 1 and Section 1 of Algebra and Bridge datasets

We aggregate the performance of all steps in the same problem to a single value and plot the aggregated performance to Figure 2. From this, we can see the sequential effect on the sequence of solving problems (from left to right). The average performance increases with the trend line, which implicitly means that forecasting methods are appropriate to cope with predicting student performance. Please note that by aggregating, we will come up with new data sets and the task now is to predict/forecast the whole problem instead of predicting/forecasting the single step in that problem. This work is, however, out of the scope of this paper, so we leave the experimental results on these new aggregated data sets for future work.

Also, in these specific data sets, the actual target variable (the actual performance) is encoded by 0 (incorrect) and 1 (correct), so we modify the equations (3) and (8) to avoid the zero value of the factor product. The Φ_{T^*k} in equation (3) now becomes:

$$\Phi_{T^*k} = \frac{\sum_{t=T^*-L}^{T^*-1} q_{tk}((p_t - 0.5) \cdot 2)}{L} \tag{10}$$

and the Φ_k in equation (8) now becomes:

$$\Phi_{T^*k} = \frac{\sum_{t=T^*-L}^{T^*-1} h'_{tk} q_{tk}((p_t - 0.5) \cdot 2)}{L} \tag{11}$$

However, other modifications on these specific data sets can also be used.

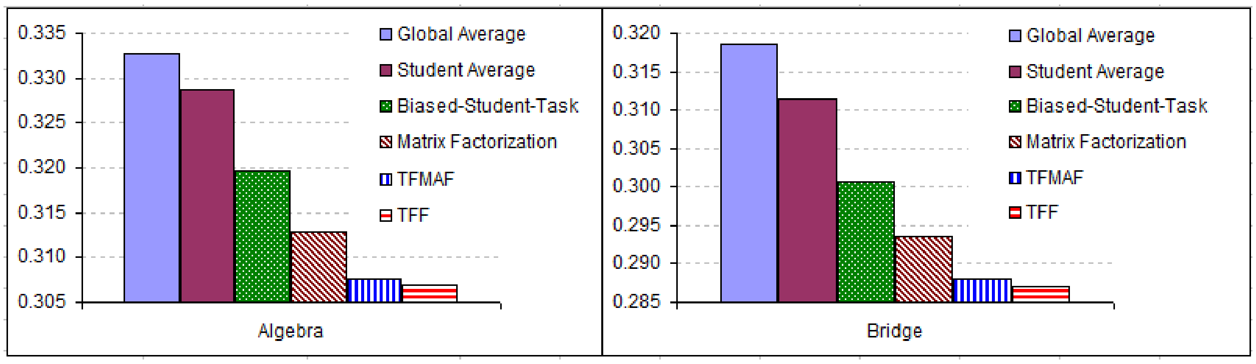

Fig. 3. RMSE results of taken into account the temporal effect using tensor factorization which factorize on student/solving-step/time.

Figure 3 presents the RMSE of the tensor factorization forecasting methods which factorize on the student (as *user*), solving-step (as *item*), and the sequence of solving-step (as *time*). The results of the proposed methods show improvement compared to the others. Moreover, compared with matrix factorization which does not take the temporal effect into account, the tensor factorization methods have also improved the prediction results. These results may implicitly reflect the natural fact that we mentioned before: "the knowledge of the student improves over time". However, the results of TFF has a small improvement compared to TFMAF method.

Table III presents the RMSE of the proposed methods and the well-known Knowledge Tracing [Corbett and Anderson 1995] which estimates the parameters by using Brute-Force (BF) [Baker et al. 2008], on Bridge data set. Since this data set is quite large, it is intractable when using Expectation Maximization (EM) method [Chang et al. 2006]. The tensor factorization forecasting models have significant improvements compared to the Knowledge Tracing model. However, the comparison with other methods, e.g. Performance Factors Analysis [Pavlik et al. 2009] and Prior Per Student [Pardos and Heffernan 2010], is leaved for future work.

Table III. RMSE of Knowledge Tracing vs. Tensor Factorization Forecasting models

Data set	Knowledge Tracing (BF)	TFMAF	TFF
Algebra	0.30561	0.30398	**0.30159**
Bridge	0.30649	0.28808	**0.28700**

For referencing, we report the hyper parameters found via cross-validation and approximation of running time in Table IV. Although the training time of TFF is high (e.g. ≈15 hours on Algebra) but in educational environment where the models need not to be retrained continuously, this running time is not an issue.

Table IV. Hyper parameters and running time. β is learning rate, λ is regularization term, K is the number of latent factors, $\#iter$ is the number of iterations, and L is the history length.

Method	Data set	Hyper parameters	Train (min.)	Test (sec.)
Matrix Factorization	Algebra	β=0.005, #iter=120, K=16, λ=0.015	16.83	0.15
TFMAF	Algebra	β=0.015, #iter=30, K=16, λ=0.015, L=8	108.84	9.17
TFF	Algebra	β=0.001, #iter=60, K=16, λ=0.015, L=10	908.71	15.11
Matrix Factorization	Bridge	β=0.01, #iter=80, K=64, λ=0.015	40.15	0.34
TFMAF	Bridge	β=0.005, #iter=20, K=64, λ=0.015, L=10	629.07	51.06
TFF	Bridge	β=0.0015, #iter=60, K=16, λ=0.005, L=5	466.01	6.61

6. DISCUSSION AND CONCLUSION

Predicting student performance is an important task in educational data mining, where we can give the students some early feedbacks to help them improving their study results. A good and reliable model which accurately predicts the student performance may replace the current standardized tests, thus, reducing the pressure on teaching and learning for examinations as well as saving a lot of time and effort for both teachers and students.

From educational point of view, the learner's knowledge improves and cumulates over time, thus, sequential effect is an important information for predicting student performance. We have proposed a novel approach - tensor factorization forecasting - which incorporates the forecasting technique into the factorization model to take into account the sequential effect.

Indeed, factorization techniques outperform other state-of-the-art collaborative filtering techniques [Koren 2010]. They belong to the family of latent factor models which aim at mapping users (students) and items (tasks) to a common latent space by representing them as vectors in that space. The performance of these techniques are promising even we do not know the background knowledge of the domain (e.g. the student/task attributes). Moreover, we use just two or three features such as student ID, task ID and/or time, thus, the memory consumption and the human effort in pre-processing can be reduced significantly while the prediction quality is reasonable. Experimental results have shown that a combination of factorization and forecasting methods can perform nicely compared to previous works which only use factorization techniques.

Another advantage of this approach is that we can personalize the prediction for each student given the task, and thus, besides predicting student performance, one could use the proposed methods to recommend the tasks (exercises) to students when building a personalized learning system.

A simple forecasting technique, which is moving average, was incorporated into the factorization model. However, applying more sophisticated forecasting techniques, e.g. Holt-Winter [Chatfield and Yar 1988; Dunlavy et al. 2011], may produce better results.

ACKNOWLEDGMENTS

The first author was funded by the "Teaching and Research Innovation Grant" project of Cantho university, Vietnam. Tomáš Horváth is also supported by the grant VEGA 1/0131/09.

REFERENCES

Baker, R. S., Corbett, A. T., and Aleven, V. 2008. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems*. Springer-Verlag, Berlin, Heidelberg, 406–415.

Baker, R. S. and Yacef, K. 2009. The state of educational data mining in 2009: A review and future visions. *Journal of Educational Data Mining (JEDM) 1,* 1, 3–17.

Bottou, L. 2004. Stochastic learning. In *Advanced Lectures on Machine Learning*, O. Bousquet and U. von Luxburg, Eds. Lecture Notes in Artificial Intelligence, LNAI 3176. Springer Verlag, Berlin, 146–168.

Brockwell, P. J. and Davis, R. A. 2002. *Introduction to Time Series and Forecasting*. Springer.

Carroll, J. and Chang, J.-J. 1970. Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition. *Psychometrika 35*, 283–319.

Cen, H., Koedinger, K., and Junker, B. 2006. Learning factors analysis a general method for cognitive model evaluation and improvement. In *Intelligent Tutoring Systems*. Vol. 4053. Springer Berlin Heidelberg, 164–175.

Cetintas, S., Si, L., Xin, Y., and Hord, C. 2010. Predicting correctness of problem solving in its with a temporal collaborative filtering approach. In *International Conference on Intelligent Tutoring Systems*. 15–24.

Chang, K., Beck, J., Mostow, J., and Corbett, A. 2006. A bayes net toolkit for student modeling in intelligent tutoring systems. In *Proceedings of International Conference on Intelligent Tutoring Systems (ITS 2006)*. Springer, 104–113.

Chatfield, C. and Yar, M. 1988. Holt-winters forecasting: Some practical issues. *Special Issue: Statistical Forecasting and Decision-Making. Journal of the Royal Statistical Society. Series D (The Statistician) 37,* 2, 129–140.

Corbett, A. T. and Anderson, J. R. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction 4*, 253–278.

Dunlavy, D. M., Kolda, T. G., and Acar, E. 2011. Temporal link prediction using matrix and tensor factorizations. *ACM Trans. Knowl. Discov. Data 5*, 10:1–10:27.

Feng, M., Heffernan, N., and Koedinger, K. 2009. Addressing the assessment challenge with an online system that tutors as it assesses. *User Modeling and User-Adapted Interaction 19,* 3, 243–266.

Gantner, Z., Drumond, L., Freudenthaler, C., Rendle, S., and Schmidt-Thieme, L. 2010. Learning attribute-to-feature mappings for cold-start recommendations. In *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM 2010).* IEEE Computer Society.

Harshman, R. A. 1970. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics 16,* 1, 84.

Koedinger, K., Baker, R., Cunningham, K., Skogsholm, A., Leber, B., and Stamper, J. 2010. A data repository for the edm community: The pslc datashop. In *Handbook of Educational Data Mining*, C. Romero, S. Ventura, M. Pechenizkiy, and R. Baker, Eds. Lecture Notes in Computer Science. CRC Press.

Kolda, T. G. and Bader, B. W. 2009. Tensor decompositions and applications. *SIAM Review 51,* 3, 455–500.

Koren, Y. 2010. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Trans. Knowl. Discov. Data 4,* 1, 1–24.

Pardos, Z. A. and Heffernan, N. T. 2010. Using hmms and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. *KDD Cup 2010: Improving Cognitive Models with Educational Data Mining*.

Pavlik, P. I., Cen, H., and Koedinger, K. R. 2009. Performance factors analysis –a new alternative to knowledge tracing. In *Proceeding of the 2009 Conference on Artificial Intelligence in Education.* IOS Press, Amsterdam, The Netherlands, 531–538.

Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web (WWW'10)*. ACM, New York, USA, 811–820.

Romero, C. and Ventura, S. 2006. *Data Mining in E-learning.* WIT Pr Computational Mechanics.

Romero, C., Ventura, S., Pechenizkiy, M., and Baker, R. S. 2010. *Handbook of Educational Data Mining.* Chapman and Hall/CRC Data Mining and Knowledge Discovery Series.

Thai-Nghe, N., Busche, A., and Schmidt-Thieme, L. 2009. Improving academic performance prediction by dealing with class imbalance. In *Proceeding of 9th IEEE International Conference on Intelligent Systems Design and Applications (ISDA'09)*. Pisa, Italy, IEEE Computer Society, 878–883.

Thai-Nghe, N., Drumond, L., Horvath, T., Krohn-Grimberghe, A., Nanopoulos, A., and Schmidt-Thieme, L. 2011. Factorization techniques for predicting student performance. In *Educational Recommender Systems and Technologies: Practices and Challenges (In press)*, O. C. Santos and J. G. Boticario, Eds. IGI Global.

Thai-Nghe, N., Drumond, L., Krohn-Grimberghe, A., and Schmidt-Thieme, L. 2010. Recommender system for predicting student performance. In *Proceedings of the 1st Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2010).* Vol. 1. Elsevier's Procedia CS, 2811 – 2819.

Thai-Nghe, N., Gantner, Z., and Schmidt-Thieme, L. 2010. Cost-sensitive learning methods for imbalanced data. *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2010)*.

Thai-Nghe, N., Horvath, T., and Schmidt-Thieme, L. 2011. Personalized forecasting student performance. In *Proceedings of the 11th IEEE International Conference on Advanced Learning Technologies (ICALT 2011) (to appear)*. IEEE CS.

Toscher, A. and Jahrer, M. 2010. Collaborative filtering applied to educational data mining. *KDD Cup 2010: Improving Cognitive Models with Educational Data Mining*.

Yu, H.-F., Lo, H.-Y., ..., and Lin, C.-J. 2010. Feature engineering and classifier ensemble for kdd cup 2010. *KDD Cup 2010: Improving Cognitive Models with Educational Data Mining*.