Multi-Relational Factorization Models for Predicting Student Performance

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

Predicting student performance (PSP) is the problem of predicting how well a student will perform on a given task. It has gained more attention from the educational data mining community recently. Previous works show that good results can be achieved by casting the PSP to rating prediction problem in recommender systems, where students, tasks and performance scores are mapped to users, items and ratings respectively. One of the most prominent approaches for rating prediction which also performs well in PSP is matrix factorization (MF). However, the state-of-the-art MF approaches for PSP only make use of one relationship, that is, between students and tasks or students and skills needed to solve the tasks. In fact each student performs several tasks, and the tasks relate to the skill(s) needed to solve them, while students are also required mastering on the skills that they have learned. In this paper we propose to exploit such multiple relationships by using multi-relational MF methods. Experiments on three large datasets show that the proposed approach can improve the prediction results.

1. INTRODUCTION

Predicting student performance (PSP) is an important task in educational data mining, where we can give the students 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 [Feng et al. 2009; Thai-Nghe et al. 2011]. Precisely, PSP 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). The benefits of PSP have been vastly discussed in the literature [Cen et al. 2006; Feng et al. 2009; Thai-Nghe et al. 2011].

To address the problem of PSP, several works have been published, e.g. as summarized in Romero et al. [2010], but most of them relied on traditional classification/regression techniques. For example, Corbett and Anderson [1995] proposed the Knowledge Tracing (KT) model, which is usually used for tracing the students' knowledge in applying their skills as well as for PSP; 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; Yu et al. [2010] used linear support vector machines together with feature engineering and ensembling techniques for predicting student performance. This approach, however, requires intensive computer memory and much human effort on data pre-processing.

Recently, researchers have proposed using recommender system techniques, e.g. k-NN collaborative filtering and matrix factorization, for PSP [Cetintas et al. 2010; Toscher and Jahrer 2010; Thai-Nghe et al. 2011]. The literature have shown that PSP can be considered as rating prediction task in recommender systems since the *student, task*, and *performance* would become *user, item*, and *rating*, respectively. The authors also shown that matrix factorization is a promising approach for PSP.

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

In fact, learning and problem-solving are complex cognitive and affective processes that are quite different to shopping and other e-commerce transactions, however, here we focus on the *"student performance"* instead of *"student preference"*. Also, as discussed in Thai-Nghe et al. [2011], the factorization models in recommender systems are able to encode latent factors of students and tasks (e.g. "slip" and "guess") implicitly, 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 has shown to be a reasonable approach. However, these published works have considered only one relationship between students and tasks.

In this work, instead of using one single relationship between students and tasks as in the literature, we propose to exploit the possible relationships between students, tasks, and their meta data for improving the prediction accuracy, using multi-relational matrix factorization (MRMF) and a weighted MRMF. This approach has shown to be successful in recommender systems [Lippert et al. 2008; Singh and Gordon 2008], however, using it in educational data mining, especially in predicting student performance is still a new topic. Our main contributions are summarized as the following:

(1) We propose a new approach for student modeling, especially for PSP, to exploit the multiple relationships between students, tasks, and their meta data by using multi-relational matrix factorization (MRMF).
(2) We also propose a weighted multi-relational matrix factorization (WMRMF) to take into account the main relation which contains the target variable.
(3) We evaluate the proposed methods on three large real-world data sets and compare their results with other state-of-the-art methods in both recommender system and student modeling domains. We empirically show that the proposed approach can improve the prediction results.

2. RELATED WORK

One of the state-of-the-art methods in PSP (or generally, student modeling) is the Knowledge Tracing (KT) [Corbett and Anderson 1995]. This model is usually used to trace the students' knowledge in applying their skills. The KT 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 the student can answer correctly even if he/she does not know the required skills for the problem; 4) slip, which is the probability that the student makes a mistake (incorrect answer) even if he/she knows the required skills. To apply the KT for predicting student performance, the four parameters need to be estimated either by using Expectation Maximization (EM) method [Chang et al. 2006] or by using Brute-Force (BF) method [Baker et al. 2008]. Recently, Pardos and Heffernan [2010] propose a variant of Knowledge Tracing by taking individualization into account.

On the other hand, in recommender system area, [Cetintas et al. 2010] proposed a temporal collaborative filtering approach to automatically predict the correctness of students' problem solving in an intelligent math tutoring system. This approach utilized the multiple interactions for a student-problem pair by using k-NN method; [Toscher and Jahrer 2010; Thai-Nghe et al. 2011] proposed using recommender system techniques (e.g. matrix factorization) for predicting student performance. The authors have shown that predicting student performance can be considered as rating prediction problem since the *student, task*, and *performance* would become *user, item*, and *rating* in recommender systems, respectively. However, these works have considered only one relationship between students and tasks.

3. PREDICTING STUDENT PERFORMANCE (PSP)

The problem of predicting student performance is to predict the likely performance of the student for some exercises (or part thereof such as for some particular steps) which we call *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 briefly summarize some concepts that will be used in this study and extend this formulation to the multi-relational case.

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

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

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

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 $\pi_{s,i}(\mathcal{D}^{test})$ (in certain cases, also given the meta data about the students and the tasks), to find

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

such that $\mathcal{E}(p, \hat{p})$ is minimal, where $p := \pi_p(\mathcal{D}^{test})$ and $\mathcal{E}$ is an error measure such as Root Mean Squared Error (RMSE).

As discussed in the literature [Toscher and Jahrer 2010; Thai-Nghe et al. 2011], matrix factorization is a good choice for PSP. In that case, however, we can use only one relationship between students and tasks, e.g. the relation "Student-Performs-Task" in Fig. 1, which can be represented as $\mathbf{R} = \{(S; I)\}$.

In this work, we would like to exploit several possible relationships between students, tasks, and their meta data, so the above formulation needs to be extended. We denote $\{\mathbf{E}_1, \ldots, \mathbf{E}_N\}$ as a set of N entity types (e.g. "Student", "Task", "Skill",...) and $\{\mathbf{R}_1, \ldots, \mathbf{R}_M\}$ as a set of M binary relation types (e.g. "Performs", "Requires",...). The problem now is to predict the values of the relation type between two entity types, e.g. $\mathbf{R}_r = \{(E_{1_r}; E_{2_r})\}$ $(r = 1 \ldots M)$, while taking into account the information in the other relations. Clearly, the multi-relational matrix factorization approach is a suitable choice for this problem.

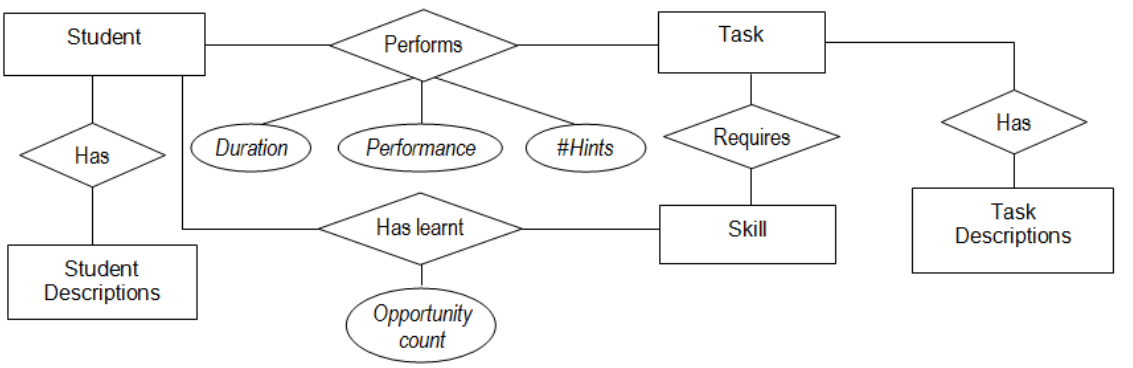

Fig. 1. Entity relationship diagram includes useful information for predicting student performance

Fig. 1 presents an example of entity relationship diagram (ERD) which covers important information in predicting student performance. Each student performs the task which is estimated by a performance score and a solving duration. The number of hints that the student requests are also expressed in this relationship. To solve the tasks correctly, the student needs to know specific skill(s), and the task itself also associates with the skill(s) that need to be learned by the students. The "opportunity count" attribute records how many times the student have learned the skill.

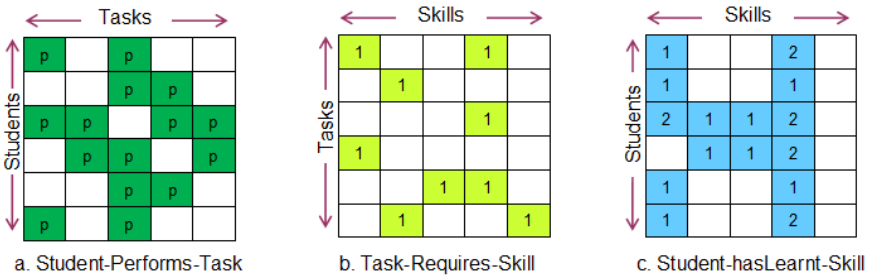

Fig. 2. An example of matrix representations (p is a performance score, e.g. $p \in [0..1]$)

Fig. 2 is an example of how to represent parts of the above ERD into matrices. The first matrix represents student performance on the given tasks (Student-Performs-Task relation); the second matrix represents whether the task requires the skills (Task-Requires-Skill relation); and the third matrix represents the number of opportunities that the student has encountered the skills (Student-HasLearnt-Skill relation).

4. METHODS

4.1 Matrix Factorization (MF)

Matrix factorization is the task of approximating[1] a matrix $\mathbf{R}$ by the product of two smaller matrices $\mathbf{W}_1$ and $\mathbf{W}_2$, i.e. $\mathbf{R} \approx \mathbf{W}_1\mathbf{W}_2^T$. $\mathbf{W}_1 \in \mathbb{R}^{S\times K}$ is a matrix where each row s is a vector containing the K latent factors describing the student s and $\mathbf{W}_2 \in \mathbb{R}^{I\times K}$ is a matrix where each row i is a vector containing the K latent factors describing the task i. Let $\mathbf{w}_{1_{sk}}$ and $\mathbf{w}_{2_{ik}}$ be the elements and $\mathbf{w}_{1_s}$ and $\mathbf{w}_{2_i}$ the vectors of $\mathbf{W}_1$ and $\mathbf{W}_2$, respectively, then the performance p given by student s to task i is predicted by:

$$\hat{p}_{si} = \sum_{k=1}^{K} \mathbf{w}_{1_{sk}}\mathbf{w}_{2_{ik}} = \mathbf{w}_{1_s}\mathbf{w}_{2_i}^T \tag{1}$$

$\mathbf{W}_1$ and $\mathbf{W}_2$ are the model parameters (latent factor matrices) which can be learned by optimizing the objective function (2) given a criterion, e.g. root mean squared error (RMSE), using stochastic gradient descent.

$$\mathcal{O}^{\text{MF}} = \sum_{(s,i)\in\mathbf{R}} \left((\mathbf{R})_{si} - \mathbf{w}_{1_s}\mathbf{w}_{2_i}^T\right)^2 + \lambda(||\mathbf{W}_1||_F^2 + ||\mathbf{W}_2||_F^2) \tag{2}$$

where $||\cdot||_F^2$ is a Frobenius norm and λ is a regularization term which is used to prevent over-fitting (please refer to the articles [Koren 2010; Thai-Nghe et al. 2011] for more details).

4.2 Multi-Relational Matrix Factorization (MRMF)

In previous section, we have briefly described the matrix factorization which uses only one relation type between two entity types (e.g. the relation "Performs" between "Student" and "Task" in Fig. 1). Multi-Relational Matrix Factorization (MRMF) [Lippert et al. 2008] is a general case of matrix factorization where we can include more than one relationship and more than two entity types.

In this study, we propose using MRMF for exploiting the multi-relational aspects in the nature of educational data, especially for predicting student performance.

[1]It has been shown that this technique works well even when $\mathbf{R}$ is very sparse [Koren 2010], which is usually the case in the PSP problem [Thai-Nghe et al. 2011]

Taking into account the multiple relationships between the entity types, the objective function of the MRMF is presented by:

$$\mathcal{O}^{\text{MRMF}} = \sum_{r=1}^{M} \sum_{(s,i)\in\mathbf{R}_r} \left((\mathbf{R}_r)_{si} - \mathbf{w}_{r_1 s}\mathbf{w}_{r_2 i}^T\right)^2 + \lambda \left(\sum_{j=1}^{N} ||\mathbf{W}_j||_F^2\right) \tag{3}$$

where M is the number of relation types and $\{\mathbf{W}_j\}_{j=1\ldots N}$ are the latent factor matrices of N entity types. Please note that equation (3) is not the sum of independent terms. When learning the model parameters, every factor matrix is updated with respect to all relation types it involves until a common convergence is met [Lippert et al. 2008] or reaching the maximum number of predefined iterations.

4.3 Weighted Multi-Relational Matrix Factorization (WMRMF)

Using MRMF, we can utilize many relationships between many entities. However, this method treats the important role of all relations equally. Clearly, we can see that the main relation which contains the target variable (e.g. "Student-Performs-Task" in Fig. 1) is more important than the other supplement relations (e.g "Task-Requires-Skill"), thus it should have more weight. We propose the Weighted Multi-Relational Matrix Factorization (WMRMF) to take into account the importance of the main relation. So, the objective function in equation (3) now becomes:

$$\mathcal{O}^{\text{WMRMF}} = \sum_{r=1}^{M} \Theta_r \sum_{(s,i)\in\mathbf{R}_r} \left((\mathbf{R}_r)_{si} - \mathbf{w}_{r_1 s}\mathbf{w}_{r_2 i}^T\right)^2 + \lambda \left(\sum_{j=1}^{N} ||\mathbf{W}_j||_F^2\right) \tag{4}$$

where Θ_r is the weight function, for example, it sets the weight to maximum for the main relation and reduces the weight for the rest, as in equation (5). However, some other choices could also be considered.

$$\Theta_r = \begin{cases} 1, & \text{if } r \text{ is the main relation} \\ \theta, & \text{else } (0 < \theta \leq 1) \end{cases} \tag{5}$$

where θ is a hyper parameter which can be determined from the training data. Another important property of the WMRMF is that in an extreme case ($\theta = 1$), the WMRMF is equivalent to the MRMF.

The WMRMF updates its latent factors for each relation at iteration n via equations (6) and (7):

$$\mathbf{w}_{r_1 s}^n = \mathbf{w}_{r_1 s}^{n-1} - \beta \left(\frac{\partial \mathcal{O}_{si}^{\text{WMRMF}}}{\partial \mathbf{w}_{r_1 s}^{n-1}}\right) \tag{6}$$

$$\mathbf{w}_{r_2 i}^n = \mathbf{w}_{r_2 i}^{n-1} - \beta \left(\frac{\partial \mathcal{O}_{si}^{\text{WMRMF}}}{\partial \mathbf{w}_{r_2 i}^{n-1}}\right) \tag{7}$$

where β is a learning rate; and the gradients $\frac{\partial \mathcal{O}_{si}^{\text{WMRMF}}}{\partial \mathbf{w}_{r_1 s}}$ and $\frac{\partial \mathcal{O}_{si}^{\text{WMRMF}}}{\partial \mathbf{w}_{r_2 i}}$ are determined by

$$\frac{\partial \mathcal{O}_{si}^{\text{WMRMF}}}{\partial \mathbf{w}_{r_1 s}} = \lambda \mathbf{w}_{r_1 s} - 2\Theta_r \left((\mathbf{R}_r)_{si} - \mathbf{w}_{r_1 s}\mathbf{w}_{r_2 i}^T\right)\mathbf{w}_{r_2 i} \tag{8}$$

$$\frac{\partial \mathcal{O}_{si}^{\text{WMRMF}}}{\partial \mathbf{w}_{r_2 i}} = \lambda \mathbf{w}_{r_2 i} - 2\Theta_r \left((\mathbf{R}_r)_{si} - \mathbf{w}_{r_1 s}\mathbf{w}_{r_2 i}^T\right)\mathbf{w}_{r_1 s} \tag{9}$$

The WMRMF's learning process is summarized in algorithm (1). We initialize the latent factor matrices from the normal distribution $\mathcal{N}(\mu, \sigma^2)$, e.g. mean $\mu = 0$ and standard deviation $\sigma^2 = 0.01$, and initialize the

weight value for each relation types. While the stopping condition is not met, e.g. reaching the maximum number of iterations or converging ($\mathcal{O}^{\text{WMRMF}}_{Iter_{(n-1)}} - \mathcal{O}^{\text{WMRMF}}_{Iter_n} < \epsilon$), the latent factors are updated iteratively.

ALGORITHM 1: LearnWMRMF($\mathbf{E}_1, \ldots, \mathbf{E}_N$: Entity types; $\mathbf{R}_1, \ldots, \mathbf{R}_M$: Relation types; λ: Regularization term; β: Learning rate; K: #Latent factors; θ: Weight value; Stopping criterion)

for $j \leftarrow 1 \ldots N$ **do**
  $\mathbf{W}_j \leftarrow$ Draw randomly from $\mathcal{N}(\mu, \sigma^2)$
end
for $r \leftarrow 1 \ldots M$ **do**
  Initialize Θ_r using equation (5)
end
while *(Stopping criterion is NOT met)* **do**
  for *each relation* $\boldsymbol{R}_r = \{(E_{1_r}; E_{2_r})\}$ *in* $\{\boldsymbol{R}_1, \ldots, \boldsymbol{R}_M\}$ **do**
    for $l \leftarrow 1 \ldots |\boldsymbol{R}_r|$, **do**
      Draw randomly (s, i) in $\mathbf{R}_r$
      $\mathbf{w}_{r_1 s} \leftarrow \mathbf{w}_{r_1 s} - \beta \left(\frac{\partial \mathcal{O}^{\text{WMRMF}}_{si}}{\partial \mathbf{w}_{r_1 s}} \right)$
      $\mathbf{w}_{r_2 i} \leftarrow \mathbf{w}_{r_2 i} - \beta \left(\frac{\partial \mathcal{O}^{\text{WMRMF}}_{si}}{\partial \mathbf{w}_{r_2 i}} \right)$
    end
  end
end
return $\{\mathbf{W}_j\}_{j=1 \ldots N}$

After the learning process, the model parameters $\{\mathbf{W}_j\}_{j=1 \ldots N}$ are obtained, then we can generate the prediction for any relation using the same equation (1).

5. EXPERIMENTS

5.1 Data sets

In the experiments described here, we use the data sets from the KDD Challenge 2010[2] [Koedinger et al. 2010] and the ASSISTments Platform[3] [Feng et al. 2009]. The original information of these data sets are described in Table I. These data represent the log files of interactions between students and the tutoring system. While students solve the 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	#Instances
Algebra-2008-2009 (Algebra)	8,918,054
Bridge-to-Algebra-2008-2009 (Bridge)	20,012,498
Assistments-2009-2010 (Assistments)	1,011,079

In the KDD 2010 data sets, namely Algebra and Bridge, 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 *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. Additionally, a different number of *skills* (or knowledge components - KCs) and associated *opportunity counts* is provided. The KCs represent specific skills used for solving the problem

[2] http://pslcdatashop.web.cmu.edu/KDDCup/
[3] http://teacherwiki.assistment.org/wiki/Assistments_2009-2010_Full_Dataset

(where available) and opportunity counts encode the number of times the respective knowledge component has been encountered before. The Assistments data set is quite similar to the above data sets. Here, we have used four attributes: Student ID, ASSISTment[4] ID, Problem ID, and the Skill.

Target of the prediction task in these data sets is the *correct first attempt* (CFA) information which encodes whether the student successfully completed the given step (or problem in Assistments data set) 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.

For Algebra and Bridge data sets, the *task* refers to a *solving-step*, which is a combination (concatenation) of *problem hierarchy*, *problem name*, *step name*, and *problem view*. For Assistments data set, the *task* refers to the *problem*. All empty values of the skill are considered as a new skill ID. Information of *student*, *task*, and *performance (CFA)* is summarized in Table II.

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

Data set	#Student	#Task	#Skill (KC)	#Performance
Algebra	3,310	1,422,200	2,979	8,918,054
Bridge	6,043	888,834	1,458	20,012,498
Assistments	8,519	35,978	348	1,011,079

ERD revisions: In these specific data sets, several information, e.g. "#Hints" and "Durations" (start time, end time), are not provided in the test sets (KDD Cup 2010 data). Thus, for applying the MRMF and WMRMF, the ERD in Fig. 1 needs to be narrow down. We propose two different ERDs for experiments as in Fig. 3. In each ERD, we also present which relation can be used as the main relation (filled by gray color), which has higher weight for the WMRMF. Moreover, the relation "Has learnt" in Fig. 1 is also revised. Instead of using "opportunity counts" as the values for this relation, we could use "average performance" of the students on the skills. By this way, we can also predict "how the students master on the given skills".

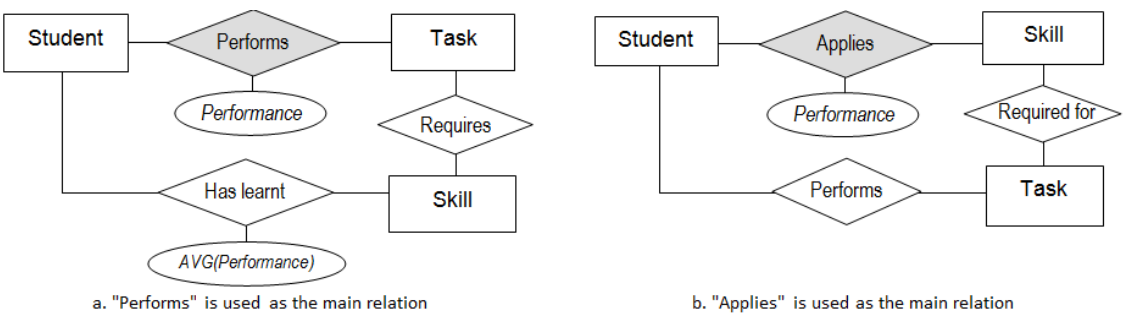

Fig. 3. ERDs are used for experiments

5.2 Baselines and experimental setting

Baselines: The proposed methods are compared with *global average*, *student average*, and *biased-student-task*[5] (this 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. 2011] shown

[4]ASSISTments are composed of questions and associated hints, solutions, web-based videos, etc.). Each ASSISTment consists of one or more problems (source: http://teacherwiki.assistment.org/wiki/About).

[5]Please see the article [Thai-Nghe et al. 2011] for details

that MF can produce competitive results. The MF also uses the same information of *user* (student), *item* (task), and *rating* (performance) as in Table II. Furthermore, the proposed methods are also compared with the state-of-the-art in student modeling - the Knowledge Tracing using Brute Force method (KT-BF) [Baker et al. 2008; Corbett and Anderson 1995].

Evaluation schema: Root mean squared error (RMSE) are used for evaluation. We would like to simulate the prediction results of the proposed methods by using a real system from KDD Challenge 2010 to see how far our models can improve compared to the others on the given data sets. Thus, the RMSE reported in this study are obtained from this website (it is still opened for submission after the challenge). Moreover, we also evaluated them on the validation sets (e.g. splitting the training set to sub-train and sub-test in the same way as described on this website).

Hyper parameter setting: Four hyper parameters of the Knowledge Tracing (prior knowledge, learn rate, slip, and guess) need to be determined. Since the Expectation Maximization (EM) method [Chang et al. 2006] runs quite slow on large data sets (even intractable on Bridge), we use the Brute-Force (BF) method [Baker et al. 2008]. The BF uses an exhaustive search to determine the hyper parameters. First, it starts a coarse search from 0.01 to 0.99 with the increment of 0.01. After the hyper parameters are found, a fine-grained search is again applied (from -0.009 to 0.009 with the increment of 0.001).

For our approach and the other baselines, the hyper parameter search is also applied (e.g., optimizing the RMSE on the validation set). However, due to the large spaces of the hyper parameters, we just did a raw search for the proposed methods, e.g. $\beta \in (10^{-4}, 10^{-3}, 10^{-2}, 5 \cdot 10^{-5}, 5 \cdot 10^{-4}, 5 \cdot 10^{-3})$, $\theta \in (0.7, 0.75, 0.8, 0.85)$, $\lambda \in (15 \cdot 10^{-4}, 15 \cdot 10^{-3}, 55 \cdot 10^{-5}, 55 \cdot 10^{-4}, 55 \cdot 10^{-3})$, $K \in (2^4, \ldots, 2^8)$. The number of iterations depend on each data set, e.g. the algorithms stop iterating when converging or over-fitting. Other choices may produce better results, though.

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 [Thai-Nghe et al. 2011].

5.3 Experimental Results

Fig. 4 presents the RMSE of the proposed methods and the other baselines (using "Student-Performs-Task" as the main relation, presented in Fig. 3a). The MRMF and WMRMF, which take into account the multiple relationships between entities, have improvements compared to the others. The WMRMF also outperforms the baseline (MF) on the validation sets (the right side of Fig. 4). These results also consist with previous work [Lippert et al. 2008] which shown that the multi-relational approach can improve over the single relational MF. Since the Bridge data set is less sparse (22.52 performances/task, on average) than the Algebra (6.27 performances/task), the factorization models perform better on Bridge.

Fig. 5 presents the RMSE results of using "Student-Applies-Skill" as the main relation (presented in Fig. 3b). In this case, instead of predicting student performance on particular task directly, we predict the student performance on the required skills associated with the task. This has been done in the literature, e.g. in [Baker et al. 2008; Corbett and Anderson 1995], for student modeling to trace how students apply their gained knowledge/skill on the given tasks. Thus, we have also experimented the KT-BF [Baker et al. 2008] on these data to understand how far our models are improved. However, it is quite expensive to obtain the KTs' hyper parameters using the Brute-Force method.

Clearly, the RMSE of the proposed methods also have improvements compared to the KT-BF. Some recent works, e.g. Performance Factors Analysis [Pavlik et al. 2009] and Prior Per Student [Pardos and Heffernan

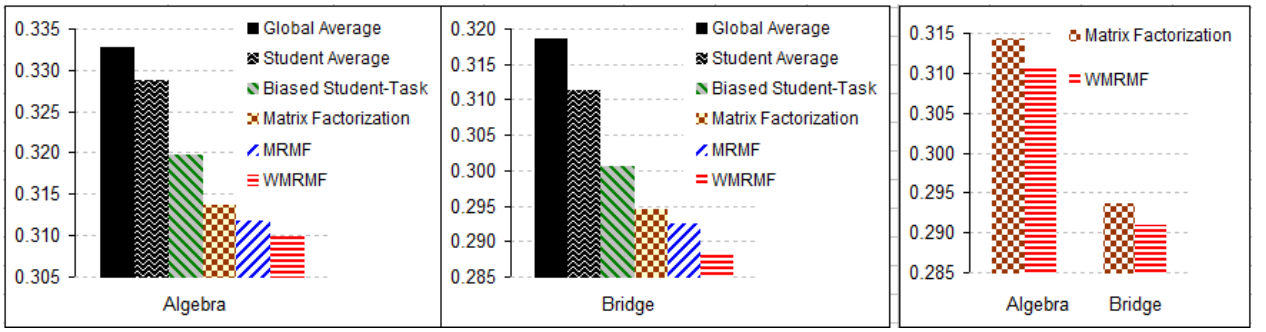

Fig. 4. RMSE of using Student-Performs-Task as the main relation (right side: RMSE on the validation sets)

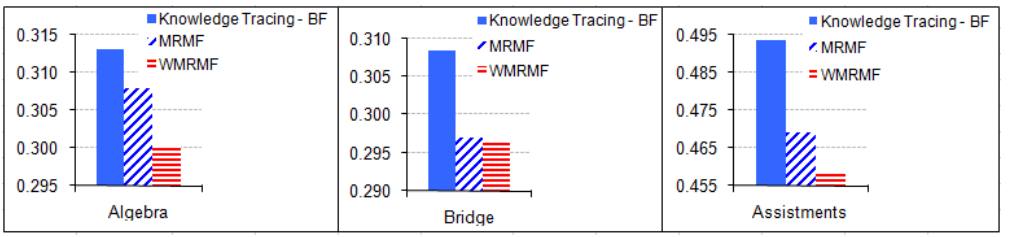

Fig. 5. Using Student-Applies-Skill as the main relation: RMSE of KT-BF vs. MRMF and WMRMF

2010], have been shown to be better performance than the KT, however, the comparison with them leaves for future work.

For referencing, we report the hyper parameters found in Table III. Running time of the WMRMF using these hyper parameters is ≈6.0 hours on the largest data set (Bridge), however, in educational environment where the models need not to be retrained continuously, this running time would not be an issue.

Table III. Hyper parameters are used for experiments

Method	Data set	Hyper parameters
Matrix Factorization (MF)	Algebra	β=0.005, #iter=120, K=16, λ=0.015
Multi-Relational Matrix Factorization (MRMF)	Algebra	β=0.0005, #iter=1000, K=16, λ=0.00055
Weighted Multi-Relational Matrix Factorization (WMRMF)	Algebra	β=0.001, #iter=550, K=16, λ=0.00125, θ=0.85
Matrix Factorization (MF)	Bridge	β=0.01, #iter=80, K=64, λ=0.015
Multi-Relational Matrix Factorization (MRMF)	Bridge	β=0.0005, #iter=700, K=40, λ=0.00055
Weighted Multi-Relational Matrix Factorization (WMRMF)	Bridge	β=0.001, #iter=550, K=80, λ=0.001, θ=0.7
Matrix Factorization (MF)	Assistments	β=0.01, #iter=80, K=64, λ=0.015
Multi-Relational Matrix Factorization (MRMF)	Assistments	β=0.005, #iter=20, K=64, λ=0.015
Weighted Multi-Relational Matrix Factorization (WMRMF)	Assistments	β=0.0015, #iter=60, K=16, λ=0.005, θ=0.7

6. CONCLUSION

We have proposed a new approach which uses multi-relational matrix factorization (MRMF) to exploit the relationships between students, tasks, and other meta data in predicting student performance. We also propose a weighted MRMF (WMRMF) to take into account the main relation that contains the target variable. We show how to present the relationships of the student performance data to multiple matrices and validate the proposed approach using three large data sets. Experimental results show that this approach can perform nicely compared to the other methods.

In future work, incorporating specific latent factors for the entities, e.g. as described in [Toscher and Jahrer 2010], and combining the results of different parameters using ensemble methods may produce better results. Moreover, adding more data relationships to the models (if applicable, e.g., the durations of performing the tasks, which highly reflect the task's difficulty; the number of hints that the student requested;...) may also lead to further improvement.

ACKNOWLEDGMENTS

The first author was funded by the TRIG project of Cantho university, Vietnam. The second author was funded by the CNPq, Brazil. Tomáš Horváth is also supported by the grant VEGA 1/0131/09. This work was funded in part by Deutsche Forschungsgemeinschaft within the project Multi-relational Factorization Models (`http://www.ismll.uni-hildesheim.de/projekte/dfg_multirel_en.html`).

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 Inter. Conf. on ITSs*. 406–415.

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.

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.

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 ICDM 2010*. IEEE Computer Society.

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 et al., Ed. CRC Press.

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

Lippert, C., Weber, S. H., Huang, Y., Tresp, V., Schubert, M., and Kriegel, H.-P. 2008. Relation-prediction in multi-relational domains using matrix-factorization. In *NIPS 2008 Workshop on Structured Input - Structured Output*. NIPS.

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.

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.

Singh, A. P. and Gordon, G. J. 2008. Relational learning via collective matrix factorization. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008)*. ACM, 650–658.

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., Horvath, T., and Schmidt-Thieme, L. 2011. Factorization models for forecasting student performance. In *Proceedings of the 4th International Conference on Educational Data Mining (EDM 2011)*.

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*.