Unified Clustering Locality Preserving Matrix Factorization for Student Performance Prediction

Chein-Shung Hwang, Yi-Ching Su

***Abstract*—Matrix factorization (MF), known as the most effective recommendation approach, has recently been used in educational contexts for predicting student performance. However, most applications neither take into account the nonnegative nature of the factor matrices nor explore the intrinsic geometric structure of the data. In this study, we propose a novel regularization framework that imposes the locality preserving constraints into the weighted regularized nonnegative MF for predicting student performance. To reduce the complexity of neighborhood evaluation, we employ the k-means clustering technique to identify groups of similar students and tasks based on the corresponding skill profiles. We also provide formal analysis for the theoretical convergence guarantees and the correctness of the iterative multiplicative updating algorithm. Experiments on two benchmark data sets demonstrate that the proposed method outperforms traditional NMF approaches and some baselines.**

***Index Terms*—matrix factorization, geometric structure, locality preserving regularization, student performance prediction.**

I. Introduction

The rapid growth of the internet and the emergence of e-commerce have led to the development of recommender systems which have emerged to solve the problem of information overload [1], [2]. The task of recommender systems is to recommend items that fit a user's taste in order to help users select items without facing an overwhelming set of choices. Recommender systems have been successfully applied in a broad range of applications, such as recommending books, movies, TV program, and music [3]-[5].

Recently, some of the techniques in recommender systems have been adopted for educational purposes. Most of this work has focused on building recommender systems to recommend objects or activities for the learners [6], [7]. On the other side, educational data mining has also been used to gain a better understanding of the student learning process and their overall involvement in it [8], [9]. One of the key tasks in educational data mining is to predict student performance, a difficult but useful task. Given such predictions, a teacher can help to focus individual student effort on potential problem areas given their performance in previous courses. In addition, curriculum committees can use prediction results to guide changes in the curriculum and to evaluate the effects of those changes [10].

Many studies have proposed to predict student performance based on various techniques, such as statistical analysis, machine learning, and data mining. Chamillard [11] used statistical analysis to predict student performance in a particular course. Observations from the analysis also provided useful insights into the relationships between courses in the curriculum. Kotsiantis et al. [12] applied five classification techniques to predict student performance in a distance learning system. It was found that the Naïve-Bayes algorithm outperformed other algorithms and achieved 74% accuracy for a two-class data set (pass/fail). Minaei-Bidgoli et al. [13] presented an approach for classifying students in order to predict their final grade based on features extracted from logged data in an education web-based system. They demonstrated that applying a genetic algorithm (GA) for optimal feature weighting could successfully improve the accuracy of combined classifier performance by about 10-12%, as compared to non-GA classifiers. Sembiring et al. [14] used Smooth Support Vector Machine classification and kernel k-means clustering techniques to predict students' final grades. Using this method, their study indicated a strong correlation between the mental condition of the students and their final academic performance.

Matrix factorization (MF), a particular type of collaborative filtering algorithm, has received much attention due to its attractive scalability and accuracy for large-scale real-world problems, such as the Netflix Challenge [15], [16]. Recently, the problem of predicting student performance has been considered a simple matrix completion problem to which researchers have applied MF techniques to the prediction of unobserved entries in sparse matrices. For example, Thai-Nghe et al. [17] applied the basic matrix factorization for predicting student performance. The authors showed that using MF techniques could improve the prediction results as compared to traditional regression methods. In a later paper, the same researchers plus two others also proposed a tensor factorization to take the temporal effect into account [18]. Toscher and Jahrerp [19] investigated the effects of several different regularization schemes. They used a neural network to blend the collected ensemble of predictions from many factor models. The model results showed that the prediction accuracy of a multi-model ensemble was considerably improved in comparison to that achieved by single models. However, these approaches took into account neither the nonnegative nature of the factor matrices nor explored the intrinsic geometric structure of the

Chein-Shung Hwang is Associate Professor at Department of Information Management, Chinese Culture University, Taiwan (cshwang@faculty.pccu.edu.tw).
Yi-Ching Su is Assistant Professor at Department of Child Care and Family Study, Asia-Pacific Institute of Creativity, Taiwan (+886-037-605771, poohh1107@gmail.com).

data.

Nonnegative MF (NMF), an extended model of MF, deals with the problem of factorizing a nonnegative matrix into two nonnegative lower-dimension factor matrices [20]. NMF is very useful in cases of complete data matrices where all entries are observed without missing values. In practice, however, the data matrix is often incomplete, as in the case of student performance prediction, where student responses are usually sparse. In this study, we incorporated ridge regression regularization to cope with the incompleteness and related overfitting problems called weighted regularized NMF (WRNMF), as described in Section 3.2.

Recent studies have shown that many real-world data can be sampled from a low-dimensional manifold that is embedded in a high-dimensional Euclidean space [21], [22]. That is, each point of the low-dimensional manifold has a neighborhood that is homeomorphic to the high-dimensional Euclidean space. To capture the geometric structure in the data, Cai et al. [23] proposed the Locality Preserving NMF (LPNMF) that uses the KL-divergence to measure the distance of two data points with added constraints between a data point and its neighbors. However, the application of LPNMF to a large amount of data usually suffers from serious computational problems.

In this study, we have proposed a novel regularization framework that imposes the locality preserving constraints into WRNMF based on the Frobenius norm minimization. We also applied our method to the problem of student performance prediction by adding constraints to both student and task neighborhoods. Due to the use of local information, our method is able to find more interpretable low-dimensional representations for students and tasks. Moreover, we employed the k-means clustering technique to identify groups of students and tasks based on the corresponding skill profile matrices, which greatly reduced the computational efforts required for pair-wise similarity computations. Concretely, the contributions of this study involved the following:

- Proposing a cluster-based locality preserving student modelling for student performance prediction.
- Providing a formal analysis for the theoretical convergence guarantees and the correctness of the iterative multiplicative updating algorithm.
- Investigating the effects of the locality preserving regularizations and comparing the proposed approaches with the standard ones.

II. Student Performance Prediction

The ability to predict student performance is very important in educational environments. Student academic performance is based upon diverse factors, such as personal, social, psychological, and other environmental variables. For this study, we were particularly interested in predicting students' ability to solve problems encountered when interacting with the computer-aided tutoring system (CATS). In CATS, a problem is typically a task a student performs that involves multiple steps and belongs to a hierarchy of units and sections. In addition, problem tasks can be described by additional information, such as the skills required to solve a specific problem. All the meta-information about tasks can be described as the attributes of the tasks. Analogously, students can be characterized by their demographics, interests, knowledge levels, etc. CATS allows for the collection of all the information about students' interactions with the tutoring system and their responses to the problem tasks. The problem of predicting student performance in CATS, thus, can be defined as estimating the response of a student for new tasks based on past performance and related meta-information stored in the system.

Let us denote the set of students by *S*, the set of tasks by *T*, and the matrix of performance scores recorded in the system by X. Suppose that no more than one score can be made by a user $s \in S$ for a particular task $t \in T$ and write this score X_{st}. Since not every student answers every task, matrix X is usually sparse. Thus, we have set $L = \{(s,t) \mid s \in S, t \in T, X_{st} \text{ is known}\}$ denoting the set of response data. Finally, we denote M_S and M_T as the set of students and tasks meta-descriptors, respectively.

Typically the set *L* is divided into a training set L_{train} and a test set L_{test}. Then the problem of predicting student performance is as follows: given L_{train}, L_{test}, M_S and M_T, to learn a function $\mathcal{F} : S \times T \to \mathbb{R}$ that predicts the score $\mathcal{F}(s, t)$ of a student s for a new task t, such that the objective function is minimal,

$$\min \sum_{(s,t) \in L_{test}} (\mathcal{F}(s,t) - X_{st})^2. \quad (1)$$

We defined the objective function as the sum of the squared error. Other error measures could, however, be used as well.

III. Matrix Factorization for Student Performance Prediction

Matrix Factorization is currently the most effective and efficient method for collaborative prediction and forms the core of many successful recommender system algorithms. The goal of MF models is to approximate a data matrix as the product of two much smaller latent matrices. MF can be described as follows: given a input matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ and a latent factor space of dimension $f, 1 \le f \le \min(m,n)$, find two matrices $\mathbf{U} \in \mathbb{R}^{m \times f}$ and $\mathbf{V} \in \mathbb{R}^{f \times n}$ whose product approximates the input matrix as closely as possible:

$$\mathbf{X} \approx \mathbf{UV} \quad (2)$$

so that UV is the low-rank approximation of X. In CATS, X represents the performance matrix describing *m* students' performance scores on *n* tasks. Each row of U contains the f latent factors describing a student's knowledge levels, and each column of V indicates the f kinds of knowledge components required for solving a task. Since each student attempts only a small portion of tasks, X is usually extremely sparse. The problem of predicting student performance can be seen as a matrix completion problem where the low-rank factors learned from observed elements are used to fill in unobserved elements of the performance matrix X.

A. Nonnegative Matrix Factorization (NMF)

The matrix factorization presented in the previous section places no constraints on the elements of factor matrices U and V. So these two matrices may contain both positive and negative elements. However, the negative elements may be counter-intuitive and difficult to interpret. For example, suppose that a student has a strong conceptual understanding of the division of fractions and thus assigned a value of 10 for that factor. If a problem task does not require this knowledge at all, it is assigned a value of -10. Multiplying these two values yields a strong negative score = -100. This means the student is more likely to fail in solving the task. However, it is possible that the student may still possess the requisite knowledge and can solve the task correctly.

Nonnegative MF is similar to MF but imposes additional constraints on the factor matrices U and V. It is required that all elements of both U and V be nonnegative. NMF aims to find two non-negative matrices whose product provides a good approximation of the original matrix. The basic NMF problem can be formulated as follows. Given the nonnegative performance matrix X, NMF aims to find two matrices $\mathbf{U} \in \mathbb{R}_+^{m \times f}$ and $\mathbf{V} \in \mathbb{R}_+^{f \times n}$ which minimize the objective function $\mathfrak{J}(\mathbf{U}, \mathbf{V})$. One popular choice of the objective function $\mathfrak{J}$ is the Euclidean distance (or the Frobenius norm):

$$\begin{aligned} &\mathfrak{J}(\mathbf{U}, \mathbf{V}) = \|\mathbf{X} - \mathbf{UV}\|_F^2 \\ &\text{subject to } \mathbf{U} \geq 0, \mathbf{V} \geq 0 \end{aligned} \tag{3}$$

A natural approach to solving this problem is to alternate between the two variables, minimizing one over one while keeping the other fixed, as proposed by Lee and Seung [20] (2001). They used an iterative multiplicative updating algorithm to minimize the objective function in Eq. (3):

$$\begin{aligned} \mathbf{U} &\leftarrow \mathbf{U} \odot \frac{(\mathbf{XV}^{\mathrm{T}})}{(\mathbf{UVV}^{\mathrm{T}})}, \\ \mathbf{V} &\leftarrow \mathbf{V} \odot \frac{(\mathbf{U}^{\mathrm{T}}\mathbf{X})}{(\mathbf{U}^{\mathrm{T}}\mathbf{UV})}, \end{aligned} \tag{4}$$

where $\odot$ and the fraction bar denote element-wise matrix product and division, respectively, and T is a matrix transpose.

Since the performance matrix X is spare, many elements are missing. We extend the NMF algorithm by incorporating binary weights into the multiplicative updates as follows [24]:

$$\begin{aligned} \mathbf{U} &\leftarrow \mathbf{U} \odot \frac{(\mathbf{W} \odot \mathbf{X})\mathbf{V}^{\mathbf{T}}}{(\mathbf{W} \odot \mathbf{UV})\mathbf{V}^{\mathbf{T}}}, \\ \mathbf{V} &\leftarrow \mathbf{V} \odot \frac{\mathbf{U}^{\mathbf{T}}(\mathbf{W} \odot \mathbf{X})}{\mathbf{U}^{\mathbf{T}}(\mathbf{W} \odot \mathbf{UV})}, \end{aligned} \tag{5}$$

where W is a binary weighting matrix, and $W_{ij} = 1$ if X_{ij} is known and $W_{ij} = 0$ otherwise. We call it WNMF, which stands for weighted NMF.

B. Weighted Regularized Nonnegative Matrix Factorization (WRNMF)

Although a learned MF model can be used to reconstruct the missing data through the product of the factor matrices, overfitting is a serious problem for large sparse datasets. This often happens when a huge number of model parameters approximate a matrix with many missing values. An illustrative example is the KDD 2010 Cup competition datasets. There are 1,146 students in the Bridge to Algebra 2006-2007 datasets with 3,656,871 total logged responses over 210,220 tasks. This means there are over 10 million free parameters for MF with $k = 50$. It is obvious that learning 10 million parameters from 3.6 million observed values will lead to overfitting.

Early stopping and regularization are two of the most common techniques to deal with the overfitting problem in MF. Usually early stopping is based on a cross-validation scheme that stops training the MF model when error on the validation set does not improve. Regularization involves introducing additional information to penalize the complexity of a learning model. In our study, the overfitting problem is avoided by adding a ridge regression regularization term [25] to the objective function, which penalizes large parameters as follow:

$$\begin{aligned} &\mathfrak{J}(\mathbf{U}, \mathbf{V}) = \|\mathbf{W} \odot (\mathbf{X} - \mathbf{UV})\|_F^2 + \lambda_{\mathrm{U}}\|\mathbf{U}\|_F^2 + \lambda_{\mathrm{V}}\|\mathbf{V}\|_F^2. \\ &\text{subject to } \mathbf{U} \geq 0, \mathbf{V} \geq 0 \end{aligned} \tag{6}$$

where $\lambda_{\mathrm{U}} \geq 0$ and $\lambda_{\mathrm{V}} \geq 0$ are the regularization parameters balancing the reconstruction error of WRNMF in the first term and regularizations in the second and third term. Then the update rules in Eq. (5) become

$$\begin{aligned} \mathbf{U} &\leftarrow \mathbf{U} \odot \frac{(\mathbf{W} \odot \mathbf{X})\mathbf{V}^{\mathrm{T}}}{(\mathbf{W} \odot \mathbf{UV})\mathbf{V}^{\mathrm{T}} + \lambda_{\mathrm{U}}\mathbf{U}}, \\ \mathbf{V} &\leftarrow \mathbf{V} \odot \frac{\mathbf{U}^{\mathrm{T}}(\mathbf{W} \odot \mathbf{X})}{\mathbf{U}^{\mathrm{T}}(\mathbf{W} \odot \mathbf{UV}) + \lambda_{\mathrm{V}}\mathbf{V}}, \end{aligned} \tag{7}$$

where $\lambda_{\mathrm{U}} \geq 0$ and $\lambda_{\mathrm{V}} \geq 0$ are the regularization parameters.

C. Locality Preserving Weighted Regularized Nonnegative Matrix Factorization (LPNMF)

In this section, the geometric structure of the performance matrix is explored. Our assumption was that if two students (or tasks) were similar, then they should be similar in their corresponding feature spaces. To apply this idea to WRNMF, we explicitly introduced two local regularization terms (one for students and the other for tasks) into the objective function in Eq. (6) to preserve the consistency of feature spaces for similar students and tasks. The closer two students or tasks are to each other in the feature spaces, the smaller the local regularizers will be. The resultant objective function is as follows:

$$\begin{aligned} \mathfrak{J}(\mathbf{U}, \mathbf{V}) &= \|\mathbf{W} \odot (\mathbf{X} - \mathbf{UV})\|_F^2 + \lambda_{\mathrm{U}}\|\mathbf{U}\|_F^2 + \lambda_{\mathrm{V}}\|\mathbf{V}\|_F^2 + \\ &\quad \beta_{\mathrm{U}} \sum_{ij} \left\| U_{i:} - U_{j:} \right\|^2 S_{ij}^{\mathrm{U}} + \beta_{\mathrm{V}} \sum_{ij} \left\| V_{:i} - V_{:j} \right\|^2 S_{ij}^{\mathrm{V}} \\ &= \|\mathbf{W} \odot (\mathbf{X} - \mathbf{UV})\|_F^2 + \lambda_{\mathrm{U}}\|\mathbf{U}\|_F^2 + \lambda_{\mathrm{V}}\|\mathbf{V}\|_F^2 + \\ &\quad \beta_{\mathrm{U}}\mathrm{tr}(\mathbf{U}^{\mathrm{T}}\mathbf{L}_{\mathrm{U}}\mathbf{U}) + \beta_{\mathrm{V}}\mathrm{tr}(\mathbf{V}\mathbf{L}_{\mathrm{V}}\mathbf{V}^{\mathrm{T}}) \\ &\quad \text{subject to } \mathbf{U} \geq 0, \mathbf{V} \geq 0 \end{aligned} \tag{8}$$

where tr() denotes the matrix trace. $\mathrm{L_U} = \mathrm{D^U} - \mathrm{S^U}$ and $\mathrm{L_V} = \mathrm{D^V} - \mathrm{S^V}$ are the Laplacian matrices for students and tasks respectively. $\mathbf{S}^{\mathrm{U}} = \left[S_{ij}^{\mathrm{U}}\right]$ and $\mathbf{S}^{\mathrm{V}} = \left[S_{ij}^{\mathrm{V}}\right]$ are the similarity matrices encoding the relationships among students and tasks, respectively. $\mathbf{D}^{\mathrm{U}} = \left[D_{ij}^{\mathrm{U}}\right]$ and $\mathbf{D}^{\mathrm{V}} = \left[D_{ij}^{\mathrm{V}}\right]$ are diagonal matrices with $D_{ii}^{\mathrm{U}} = \sum_j S_{ij}^{\mathrm{U}}$ and $D_{ii}^{\mathrm{V}} = \sum_j S_{ij}^{\mathrm{V}}$. Finally, β_{U} and β_{V} are the two positive parameters controlling the contributions of the locality preserving regularizations. The

crucial part of local regularizations is the definition of the similarity matrices $\mathbf{S}^{\mathrm{U}}$ and $\mathbf{S}^{\mathrm{V}}$, which will be explored in the following subsections.

The objective function $\mathfrak{J}$ of LPNMF in Eq. (8) is not convex for U and V simultaneously, but it is convex for U when V is fixed and it is also convex for V when U is fixed. Following the work of Lee and Seung [20], we present an alternating scheme to optimize the objective.

Since $\mathbf{U} \geq 0$ and $\mathbf{V} \geq 0$, we introduce two Lagrangian multipliers, $\mathbf{\Phi} \in \mathbb{R}^{m \times f}$ and $\mathbf{\Psi} \in \mathbb{R}^{f \times n}$; thus the Lagrangian function $\mathcal{L}$ is

$$\begin{aligned}\mathcal{L}(\mathbf{U},\mathbf{V}) = \; & \|\mathbf{W} \odot (\mathbf{X} - \mathbf{UV})\|_F^2 + \lambda_{\mathrm{U}}\|\mathbf{U}\|_F^2 + \lambda_{\mathrm{V}}\|\mathbf{V}\|_F^2 + \\ & \beta_{\mathrm{U}}\mathrm{tr}\left(\mathbf{U}^{\mathrm{T}}\mathbf{L}_{\mathrm{U}}\mathbf{U}\right) + \beta_{\mathrm{V}}\mathrm{tr}\left(\mathbf{V}\mathbf{L}_{\mathrm{V}}\mathbf{V}^{\mathrm{T}}\right) - \mathrm{tr}\left(\mathbf{\Phi}\mathbf{U}^{\mathrm{T}}\right) - \mathrm{tr}\left(\mathbf{\Psi}\mathbf{V}^{\mathrm{T}}\right)\end{aligned} \tag{9}$$

The partial derivatives of $\mathcal{L}$ with, respectively, U and V are

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{U}} = \; & -2(\mathbf{W} \odot \mathbf{X})\mathbf{V}^{\mathrm{T}} + 2(\mathbf{W} \odot \mathbf{UV})\mathbf{V}^{\mathrm{T}} \\ & + 2\lambda_{\mathrm{U}}\mathbf{U} + 2\beta_{\mathrm{U}}\mathbf{L}_{\mathrm{U}}\mathbf{U} - \mathbf{\Phi} \\ \frac{\partial \mathcal{L}}{\partial \mathbf{V}} = \; & -2\mathbf{U}^{\mathrm{T}}(\mathbf{W} \odot \mathbf{X}) + 2\mathbf{U}^{\mathrm{T}}(\mathbf{W} \odot \mathbf{UV}) \\ & + 2\lambda_{\mathrm{V}}\mathbf{V} + 2\beta_{\mathrm{V}}\mathbf{V}\mathbf{L}_{\mathrm{V}} - \mathbf{\Psi}\end{aligned} \tag{10}$$

Using the Karush-Kuhn-Tucker complementary condition $\mathbf{\Phi} \odot \mathbf{U} = 0$ and $\mathbf{\Psi} \odot \mathbf{V} = 0$, we get

$$\begin{aligned}& (-2(\mathrm{W} \odot \mathrm{X})\mathrm{V}^{\mathrm{T}} + 2(\mathrm{W} \odot \mathrm{UV})\mathrm{V}^{\mathrm{T}} \\ & \quad + 2\lambda_{\mathrm{U}}\mathbf{U} + 2\beta_{\mathrm{U}}\mathbf{L}_{\mathrm{U}}\mathbf{U}) \odot \mathbf{U} = 0 \\ & (-2\mathbf{U}^{\mathrm{T}}(\mathbf{W} \odot \mathbf{X}) + 2\mathbf{U}^{\mathrm{T}}(\mathbf{W} \odot \mathbf{UV}) \\ & \quad + 2\lambda_{\mathrm{V}}\mathbf{V} + 2\beta_{\mathrm{V}}\mathbf{V}\mathbf{L}_{\mathrm{V}}) \odot \mathbf{V} = 0\end{aligned} \tag{11}$$

Equation (11) leads to the following update rules:

$$\begin{aligned}\mathbf{U} &\leftarrow \mathbf{U} \odot \frac{(\mathbf{W} \odot \mathbf{X})\mathbf{V}^{\mathrm{T}} + \beta_{\mathrm{U}}\mathbf{S}^{\mathrm{U}}\mathbf{U}}{(\mathbf{W} \odot \mathbf{UV})\mathbf{V}^{\mathrm{T}} + \lambda_{\mathrm{U}}\mathbf{U} + \beta_{\mathrm{U}}\mathbf{D}^{\mathrm{U}}\mathbf{U}}, \\ \mathbf{V} &\leftarrow \mathbf{V} \odot \frac{\mathbf{U}^{\mathrm{T}}(\mathbf{W} \odot \mathbf{X}) + \beta_{\mathrm{V}}\mathbf{V}\mathbf{S}^{\mathrm{V}}}{\mathbf{U}^{\mathrm{T}}(\mathbf{W} \odot \mathbf{UV}) + \lambda_{\mathrm{V}}\mathbf{V} + \beta_{\mathrm{V}}\mathbf{V}\mathbf{D}^{\mathrm{V}}},\end{aligned} \tag{12}$$

D. Convergence Analysis

In this section, we make use of an auxiliary function similar to that used in Lee and Seung [20] to prove the convergence of the update rules in Eq. (12). Here we only treated the update rule for U since that of V can be proved in a similar fashion. We first introduce the definition of an auxiliary function.

Definition 3.1. $Z(U, U')$ is an auxiliary function for F(U) if the conditions

$$Z(U, U') \geq F(U), Z(U, U) = F(U) \tag{13}$$

are satisfied.

Lemma 3.2. If Z is an auxiliary function for F, then F is nonincreasing under the update [20]

$$U^{(t+1)} = \arg\min_{U} Z(U, U^{(t)}) \tag{14}$$

Proof:

$$F(U^{(t+1)}) \leq Z(U^{(t+1)}, U^{(t)}) \leq Z(U^{(t)}, U^{(t)}) = F(U^{(t)})$$

We now show that by defining an appropriate auxiliary function, the update rule for U in Eq. (12) easily follows from Eq. (14).

For any element U_{ij} of U, optimizing $\mathfrak{J}(\mathbf{U}, \mathbf{V})$ with respect to U_{ij} is equivalent to optimizing

$$\begin{aligned}\mathfrak{J}(\mathbf{U}) = \; & \|\mathbf{W} \odot (\mathbf{X} - \mathbf{UV})\|_F^2 + \lambda_{\mathrm{U}}\|\mathbf{U}\|_F^2 + \beta_{\mathrm{U}}\sum_{ij}\|U_{i:} - U_{j:}\|^2 S_{ij}^{\mathrm{U}} \\ & \text{subject to } \mathbf{U} \geq 0\end{aligned} \tag{15}$$

Let F_{ij} denote the part of $\mathfrak{J}(\mathbf{U})$ which is only relevant to U_{ij}. The first and the second-order derivatives of F_{ij} are

$$\begin{aligned}F'_{ij} &= \begin{pmatrix} -2(\mathbf{W} \odot \mathbf{X})\mathbf{V}^{\mathrm{T}} \\ + 2(\mathbf{W} \odot \mathbf{UV})\mathbf{V}^{\mathrm{T}} + 2\lambda_{\mathrm{U}}\mathbf{U} + 2\beta_{\mathrm{U}}\mathbf{L}_{\mathrm{U}}\mathbf{U} \end{pmatrix}_{ij} \\ F''_{ij} &= 2\left(W_{i:} \odot V_{j:}\right)V_{j:}^{\mathrm{T}} + 2\lambda_{\mathrm{U}} + 2\beta_{\mathrm{U}}(\mathbf{L}_{\mathrm{U}})_{ij}\end{aligned} \tag{16}$$

To prove that F_{ij} is nonincreasing under the update rules of Eq. (12), the following auxiliary function is introduced for F_{ij}.

Lemma 3.3. *Function*

$$\begin{aligned}Z(U, U_{ij}^{(t)}) = \; & F_{ij}(U_{ij}^{(t)}) + F'_{ij}(U_{ij}^{(t)})(U - U_{ij}^{(t)}) + \\ & \frac{((\mathbf{W} \odot \mathbf{UV})\mathbf{V}^{\mathrm{T}} + \lambda_{\mathrm{U}}\mathbf{U} + \beta_{\mathrm{U}}\mathbf{D}^{\mathrm{U}}\mathbf{U})_{ij}}{U_{ij}^{(t)}}(U - U_{ij}^{(t)})^2\end{aligned} \tag{17}$$

is an auxiliary function for F_{ij}.

Proof: Since $Z(U, U) = F_{ij}(U)$ is obvious, we need only show that $Z(U, U_{ij}^{(t)}) \geq F_{ij}(U)$. To do this, we compare the Taylor series expansion of $F_{ij}(U)$

$$\begin{aligned}F_{ij}(U) = \; & F_{ij}(U_{ij}^{(t)}) + F'_{ij}(U_{ij}^{(t)})(U - U_{ij}^{(t)}) + \\ & \left[\left(W_{i:} \odot V_{j:}\right)V_{j:}^{\mathrm{T}} + \lambda_{\mathrm{U}} + \beta_{\mathrm{U}}(\mathbf{L}_{\mathrm{U}})_{ij}\right](U - U_{ij}^{(t)})^2\end{aligned} \tag{18}$$

with Eq. (17) to find that $Z(U, U_{ij}^{(t)}) \geq F_{ij}(U)$ is equivalent to

$$\begin{aligned}& \frac{((\mathbf{W} \odot \mathbf{UV})\mathbf{V}^{\mathrm{T}} + \lambda_{\mathrm{U}}\mathbf{U} + \beta_{\mathrm{U}}\mathbf{D}^{\mathrm{U}}\mathbf{U})_{ij}}{U_{ij}^{(t)}} \geq \\ & \left(W_{i:} \odot V_{j:}\right)V_{j:}^{\mathrm{T}} + \lambda_{\mathrm{U}} + \beta_{\mathrm{U}}(\mathbf{L}_{\mathrm{U}})_{ii}\end{aligned} \tag{19}$$

Since we have

$$\begin{aligned}& ((\mathbf{W} \odot \mathbf{UV})\mathbf{V}^{\mathrm{T}})_{ij} = \\ & \sum_{c=1}^{n} W_{ic}\left(\sum_{l=1}^{f} U_{il}^{(t)}V_{lc}^{(t)}\right)V_{jc}^{(t)} \geq \\ & \sum_{c=1}^{n} W_{ic}U_{ij}^{(t)}V_{jc}^{(t)}V_{jc}^{(t)} = \left(W_{i:} \odot V_{j:}\right)V_{j:}^{\mathrm{T}}U_{ij}^{(t)}\end{aligned} \tag{20}$$

and

$$\begin{aligned}& (\mathbf{D}^{\mathrm{U}}\mathbf{U})_{ij} = \\ & \sum_{c=1}^{m} D_{ic}^{U}U_{cj}^{(t)} \geq D_{ii}^{U}U_{ij}^{(t)} \geq \left(D^{U} - S^{U}\right)_{ii}U_{ij}^{(t)} = (\mathbf{L}_{\mathrm{U}})_{ii}U_{ij}^{(t)}\end{aligned} \tag{21}$$

thus, Eq. (17) holds and $Z(U, U_{ij}^{(t)}) \geq F_{ij}(U)$.

According to Lemma 3.2, the update rule for $F_{ij}(U)$ can be solved by setting the gradient of $Z(U, U_{ij}^{(t)})$ to zero:

$$\begin{aligned}& \frac{\partial Z(U, U_{ij}^{(t)})}{\partial U} = F'_{ij}(U_{ij}^{(t)}) + \\ & 2\frac{((\mathbf{W} \odot \mathbf{UV})\mathbf{V}^{\mathrm{T}} + \lambda_{\mathrm{U}}\mathbf{U} + \beta_{\mathrm{U}}\mathbf{D}^{\mathrm{U}}\mathbf{U})_{ij}}{U_{ij}^{(t)}}(U - U_{ij}^{(t)}) = 0\end{aligned} \tag{22}$$

Now we can derive the same update rule as that in Eq. (12):

$$U_{ij}^{(t+1)} = U_{ij}^{(t)} - U_{ij}^{(t)} \frac{F'_{ij}(U_{ij}^{(t)})}{2((\mathbf{W} \odot \mathbf{UV})\mathbf{V}^{\mathrm{T}} + \lambda_{\mathrm{U}}\mathbf{U} + \beta_{\mathrm{U}}\mathbf{D}^{\mathrm{U}}\mathbf{U})_{ij}} = U_{ij}^{(t)} \frac{\left((\mathbf{W} \odot \mathbf{X})\mathbf{V}^{\mathrm{T}} + \beta_{\mathrm{U}}\mathbf{S}^{\mathrm{U}}\mathbf{U}\right)_{ij}}{((\mathbf{W} \odot \mathbf{UV})\mathbf{V}^{\mathrm{T}} + \lambda_{\mathrm{U}}\mathbf{U} + \beta_{\mathrm{U}}\mathbf{D}^{\mathrm{U}}\mathbf{U})_{ij}} \quad (23)$$

E. Similarity Computation

The crucial part of the LPNMF model is computing the similarity matrices $\mathbf{S}^{\mathrm{U}}$ and $\mathbf{S}^{\mathrm{V}}$ for students and tasks, respectively. Various types of information can be encoded to define the proper similarity matrices. The most direct and intuitive way is using the student performances based on the assumptions that similar students have similar responses to the tasks and similar tasks likely receive similar student responses. However, this model is usually limited by the need to compute all pair-wise similarities between students and tasks due to the large amount of input data. The computational complexities of the similarity calculation are $O(m^2n)$ for students and $O(n^2m)$ for tasks.

Instead of using the information from a single performance matrix, in this study we incorporated the skill information into the student performance prediction problem. We assumed that two students with similar skills would perform similarly and that two tasks with similar skill requirements would be given similar responses by the students. In CATS, each task is associated with zero or more skills representing the knowledge required for its solution. The skill dependencies of each task form a matrix called the Q-matrix [26]. The matrix $\mathbf{Q} \in \mathbb{R}^{n \times q}$ is usually an expert-generated matrix where $Q_{jl} = 1$ if task j requires skill l and equals 0 if it does not. Besides, a performance matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ is also defined where $X_{ij} = 1$ if student i answers task j correctly and equals 0 if he answers incorrectly. If student i did not answer task j, then $X_{ij} = NA$.

In this study, we adopted the capability matrix proposed by Ayers et al. [27] to estimate the student skill profile. Capability matrix C is a $m \times q$ matrix where C_{il} is the proportion of correctly answered tasks involving skill l that student i attempted. That is

$$C_{il} = \frac{\sum_{j=1}^{n} W_{ij}X_{ij}Q_{jl}}{\sum_{j=1}^{n} W_{ij}Q_{jl}} \quad (26)$$

where W is the binary weighting matrix defined before. The capability matrix expands on sum-score by accounting for the number of tasks requiring skill l that student i answered. Values of C_{il} indicate the degree of certainty about skill mastery. For each C_{il}, zero indicates no skill mastery, one means complete mastery, and values in between are less certain. If a student has not seen any of the tasks requiring a particular skill, a value of 0.5 is assigned, indicating uncertainty. It can be seen that using the Q-matrix and the capability matrix, the computational complexities for the similarity computation can be reduced to $O(m^2q)$ for students and $O(n^2q)$ for tasks, where $q << \min(m,n)$. However, the exponential complexity still renders it impractical for a large scale dataset.

In response, we applied the *k*-means clustering method to the capability matrix and the Q-matrix to identify groups of students and tasks of similar skill profiles. Different distance/similarity measures were used for different types of data. We used the Euclidean distance for the capability matrix and the cosine similarity for the asymmetric binary Q-matrix. A task is not assigned to any groups if it does not require any skills to complete. The similarities between two users and two tasks are thus defined as

$$S_{ij}^{\mathrm{U}} = \begin{cases} 1 & \text{if } g^{\mathrm{U}}(i) = g^{\mathrm{U}}(j) \\ 0 & \text{otherwise} \end{cases} \qquad S_{ij}^{\mathrm{V}} = \begin{cases} 1 & \text{if } g^{\mathrm{V}}(i) = g^{\mathrm{V}}(j) \\ 0 & \text{otherwise} \end{cases} \quad (25)$$

where $g^{\mathrm{U}}(i)$ and $g^{\mathrm{V}}(j)$ give the cluster number for student i and task j, respectively. The similarity measures assign a value of 1 if two students or tasks are in the same group and a value of 0 if they are in different groups. The assignment of binary values imposes penalty on the local regularizations for students or tasks that are similar in the performance spaces but different in the feature spaces. The *k*-means method provides an efficient way to do similarity computation that reduces the time complexity to $O(kmq)$ for students and $O(knq)$ for tasks. We refer to the model LPNMF with clustering on both students and tasks as a Unified Clustering LPNMF.

IV. Experiments

In this section, the experimental results are reported to evaluate the performance of the proposed method. Specifically, we investigated the effects of the clustering and locality preserving regularizations on student performance prediction. We also made comparisons with other baselines, such as the global average and the student average.

A. Datasets and Evaluation Metrics

The data were from the 2010 KDD Cup competition on educational data mining. We used the Algebra 2006-2007 and the Bridge to Algebra 2006-2007 datasets. In the rest of this paper, they are referred to as "Algebra" and "Bridge", respectively. Each data set contained the log files of interactions between students and CATS and was split into training and test partitions as described in Table I.

TABLE I
DATA SETS

Data sets	Students	Steps	Tasks	Skills
Algebra (training)	1,338	2,270,384	590,672	491
Algebra (test)	502(new)	19,342	4,697	222(new)
Bridge (training)	1,146	3,649,199	209,802	493
Bridge (test)	0(new)	7,672	418	0(new)

The central element of interaction between students and CATS is the *problem*. Each problem consists of many sub questions called *steps* and belongs to a hierarchy of *unit* and *section*. A combination of problem hierarchy, problem name, and step name forms a solving step called *tasks*. Additionally, each task is associated with some *knowledge components*, relevant concepts or skills that are required to perform that task correctly. Finally, student responses to the problems are

encoded as *correct first attempt* (CFA) in the data sets. The task of the KDD competition was to predict the CFA for each test data.

The performance of the MF algorithms was evaluated by the root mean square error (RMSE) measure, which is defined as follows:

$$\text{RMSE}=\sqrt{\frac{\sum_{(i,j)\in L_{test}}\left(X_{ij}-\widehat{X}_{ij}\right)^2}{|L_{test}|}} \quad (27)$$

where L_{test} represents the set of test data, X_{ij} is the actual response value for task *j* by student *i*, and $\widehat{X}_{ij}$ is the predicted CFA value by the MF models.

B. Model Settings

Several parameters affect the performance of the MF algorithms. In the following experiments, the ridge regression parameters λ_U and λ_V were set as equal and tuned by searching the grid {0.001, 0.005, 0.01, 0.05, 0.1, 0.5}. The dimensionality of latent factors *f* was set by the grid {2, 4, 8, 16, 32, 64}. In GRNMF, the two locality preserving regularizations β_U and β_V were set by searching the grid {0.001, 0.01, 0.1, 1, 10, 50, 100}. Finally, the number of clusters in the *k*-means algorithm was set to a range of 1 to 70 with increments of 10.

For each data set, we used the training set to build the student and the task factor matrices and then predicted the test set by the product of these two matrices. As can be seen in the Algebra data set, some new test data (student and task) were not included in the training set. To alleviate this so-called cold start problem, we provided the global average score for the new students or new tasks.

C. Performance on Cluster Size

One of main tasks of the cluster-based LPNMF approach is to group similar students or tasks using the *k*-means algorithm. The larger the cluster size, the less similar the students or tasks in the corresponding cluster. On the other hand, small cluster sizes imply fewer students or tasks involved in the locality preserving regularization. To evaluate the sensitivity of different cluster sizes, we empirically set $f = 32$, $\lambda_U = \lambda_V = 0.01$, $\beta_U = 50$, and $\beta_V = 0.01$. Fig. 1 illustrates the effects of the cluster size on RMSE for student clustering and task clustering LPNMF. In both approaches, the prediction quality improved as the number of clusters increased until an optimal point was reached; any further increments failed to improve or even gave worse results. The optimal numbers of clusters were 30 and 60 for student clustering and task clustering LPNMF in the Bridge data set and 20 and 50 in the Algebra data set, respectively.

D. Comparison

In the next experiment, we ran each approach separately with different combinations of parameter values for the Algebra and Bridge data sets and selected the best performance from each approach for comparisons. Fig. 2 presents the comparison of RMSE for the different approaches. The cluster-based LPNMF approaches provided significantly better performance than traditional regularized NMF and baseline methods. This showed that by considering the intrinsic geometrical structure of the data, LPNMF could learn a better compact representation of the knowledge structures. For both cluster-based approaches, the task clustering LPNMF performed better than the student clustering LPNMF on both data sets; this implied that clustering over the Q-matrix could better capture the geometric structure than clustering over the capability matrix. Overall, the unified clustering LPNMF performed the best in all cases.

E. Impact of Parameters β_U and β_V

The choice of regularization parameters has a significant impact on the performance of an algorithm. The determination of good parameters is usually tedious and data dependent. To investigate the impact of parameters β_U and β_V on the performance of the corresponding approaches, we set the number of clusters to the optimal numbers suggested from previous experiments. We also fixed the dimensions of both student and task factors to 32. Fig. 3 shows how the RMSE changed with respect to varying parameter values. The parameter values had significant influences on the performance, but these influences could be quite different and move in opposite directions. The performance of the student clustering LPNMF became worse if parameter β_U was set to a small value. In contrast, the task clustering LPNMF had a better performance when β_V was small. However, both parameters β_U and β_V reached their minima at 50 and 0.01, respectively.

We observed a considerable difference in the optimal values between β_U and β_V. One possible reason for this was the unbalanced dimensions in the performance matrix. In the case of the Bridge datasets, the number of tasks was over 200,000: that is about 200 times the number of students. That meant that many more tasks than students were involved in the locality preserving regularization. Therefore, the value of β_V should be small enough to avoid over-regularizing.

F. Efficiency of Analysis

The complexity analysis in the previous section indicates that the computational complexity of our approach is linear with respect to the size of task and student clusters, which proves that our approach is scalable to very large datasets. To demonstrate the performance efficiency, we ran the Unified Clustering LPNMF using the best parameter values via cross validation. In experiments of the Bridge and the Algebra data sets, each iteration needed less than 3 seconds. In Fig. 4, we plot both training and test performance measured by RMSE with respect to the number of iterations. In both experiments, our proposed method needs less than 100 iterations to converge, which only takes approximately 5 minutes.

V. CONCLUSION

The intrinsic geometric structure in the data provides an important source of information regarding data points and their neighborhood. This is especially useful for student performance prediction. In this study, we proposed a unified clustering locality preserving NMF which made predictions by considering students' performance histories, skill profiles and influences from neighbors. In particular, we constructed two clusters on students as well as tasks based on their skill profiles. These clusters represented the neighborhood

structure of the corresponding data, which greatly reduced the complexity of neighborhood evaluation. To investigate the effects of the clustering and locality preserving regularizations, we compared the proposed approaches with the traditional NMF and some baselines. In terms of the prediction accuracy, the cluster-based LPNMF approaches achieved better results. They yielded 5% and 12% improvements on average, compared to that of WRNMF and baselines, respectively. The experimental results confirmed the effects of intrinsic geometric structure on prediction accuracy. Further, the performance of the cluster-based LPNMF can be improved by incorporating both student and task regularizations. Overall, the unified clustering LPNMF performed the best in all cases.

Our study measured students' knowledge using whole history information, which assumed nothing was forgotten. Student knowledge, however, is not static and changes over the time; thus, the temporal effect is an important factor in predicting student performance. Our future works will include incorporating temporal information into the LPNMF. Several approaches, such as tensor factorization [18] and feature-based MF [28], will be investigated.

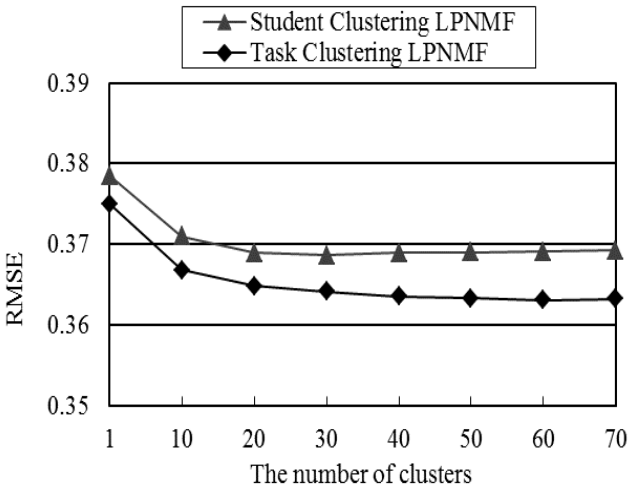

Fig. 1a. Impact of the number of clusters on Algebra data sets

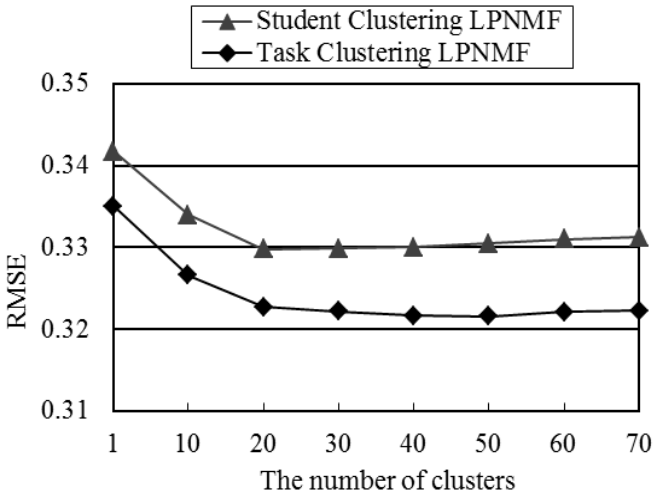

Fig. 1b. Impact of the number of clusters on Bridge data sets

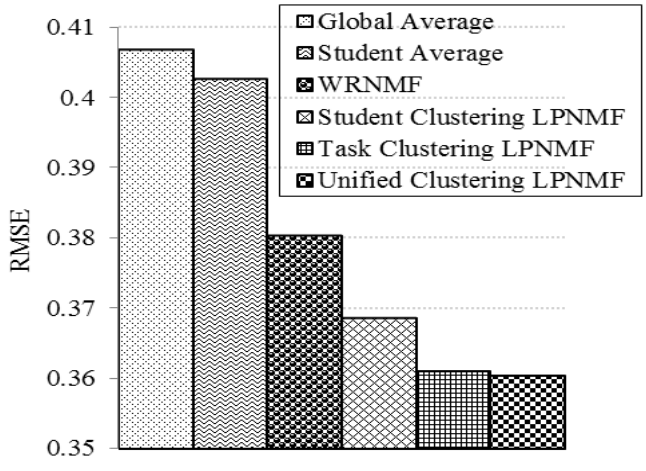

Fig. 2a. RMSE results for different methods on Algebra data sets

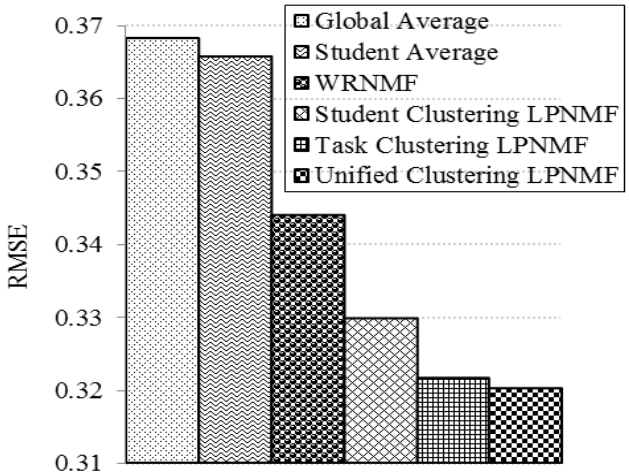

Fig. 2b. RMSE results for different methods on Bridge data sets

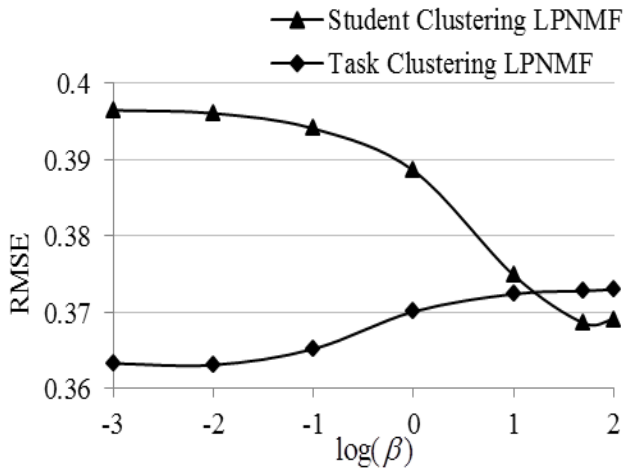

Fig. 3a. Impact of β_U and β_V on Algebra data sets

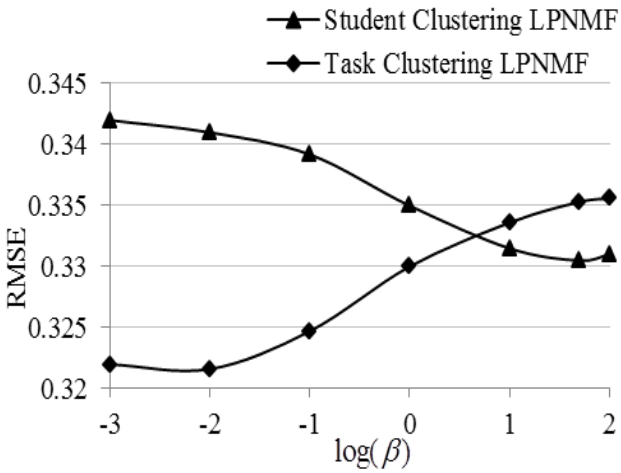

Fig. 3b. Impact of β_U and β_V on Bridge data set

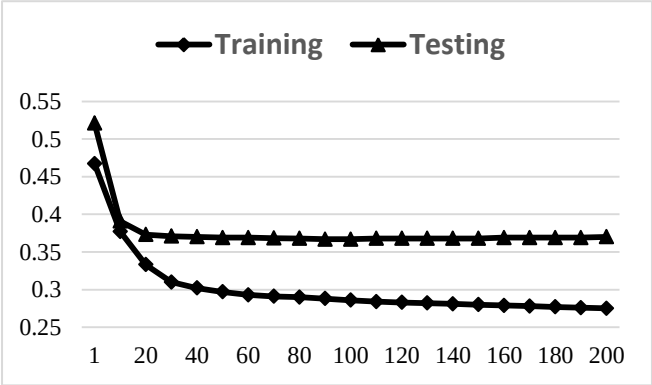

Fig. 4a. Training and Test performance on Algebra data sets

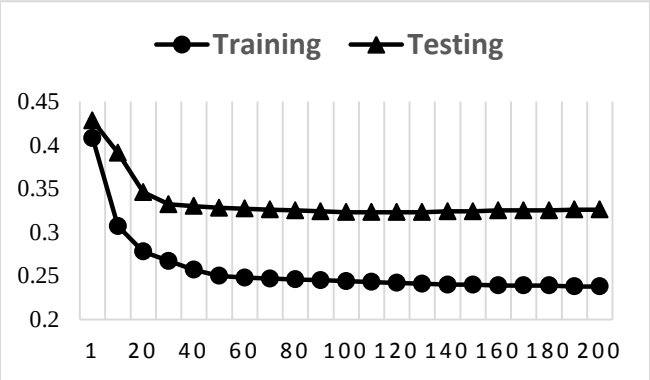

Fig. 4b. Training and Test performance on Bridge data sets

REFERENCES

[1] R. Burke, "Hybrid recommender systems: survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331-370, 2002.

[2] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems:a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[3] B. Smyth, and P. Cotter, "A personalized television listings service," *Communications of The ACM*, vol. 43, no.8 , pp. 107-111, 2000.

[4] Y. I. Chang, C. C. Wu and M. C. Tsai, "A Fair Approach to Music Recommendation Systems Based on Music Data Grouping," IAENG International Journal of Computer Science, vol. 38, no. 4, pp. 418-427, 2011.

[5] J. Kim, K. Kim, K. H. You, and J. H. Lee, "An approach for music recommendation using content-based analysis and collaborative filtering," *Information: An International Interdisciplinary Journal*, vol. 15, no. 5, pp. 1985-1996, 2012.

[6] K. Ghauth, and N. Abdullah, "Learning materials recommendation using good learners' ratings and content-based filtering," *Educational Technology Research and Development*, vol. 58, no 6, pp. 711-727, 2010.

[7] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, and R. Koper, "Recommender systems in technology enhanced learning," in *Recommender systems handbook*, 1st ed. Kantor, P.B., Ricci, F., Rokach, L., Shapira, B. (eds.). New York: Springer, 2011, pp. 387-415.

[8] M. Saarela and T. Kärkkäinen, "Analysing Student Performance using Sparse Data of Core Bachelor Courses," *JEDM - Journal of Educational Data Mining*, vol. 7, no. 1, pp. 3-32, 2015.

[9] M. Falakmasir, and J. Habibi, "Using educational data mining methods to study the impact of virtual classroom in e-Learning," In *Proc. 3rd International Conference on Educational Data Mining*, Pittsburgh, PA, USA, 2010, pp. 241-248.

[10] A. Herhskovitz, and R. Nachmias, "Learning about online learning processes and students' motivation through Web usage mining," *Interdisciplinary Journal of E-Learning and Learning Objects*, vol. 5, pp. 197-214, 2009.

[11] A. T. Chamillard, "Using student performance predictions in a computer science curriculum," In *Proc. 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, ACM, Bologna, Italy, 2006, pp. 260-264.

[12] S. Kotsiantis, C. Pierrakeas, and P. Pintelas, "Predicting of students performance in distance learning using machine learning techniques." *Applied Artificial Intelligence*, vol. 18, no. 5, pp. 411-426, 2004.

[13] B. Minaei-bidgoli, D.A. Kashy, G. Kortmeyer, and W. F. Punch, "Predicting student performance: An application of data mining methods with the educational Web-based system LON-CAPA," In *Proc. The ASEE/IEEE International Conference on Frontiers in Education*, Boulder, 2003, pp. 13-18.

[14] S. Sembiring, M. Zarlis, D. Hartama, Ramliana and E. Wani, "Prediction of student academic performance by an application of data mining techniques," In *Proc. International Conference on Management and Artificial Intelligence*, IACSIT Press, Bali, Indonesia, 2011, pp. 110-114

[15] G. Takács, I. Pilászy, B. Németh, and D. Tikk, "Matrix factorization and neighbor based algoritms for the Netflix Prize problem," in *Proc. ACM Conference on Recommender Systems*, Lausanne, Switzerland, 2008, pp. 267–274.

[16] Q. Zheng, C. K. Chan and H. D. S. IP, "IURA: An Improved User-based Collaborative Filtering Method Based on Innovators ," in *Proc. International MultiConference of Engineers and Computer Scientists 2014 Vol I, IMECS 2014*, Hong Kong, 2014, pp.366-371.

[17] N. Thai-Nghe, L. Drumond, A. Krohn-Grimberghe, and L. Schmidt-Thieme, "Recommender system for predicting student performance," in *Proc. The 1st Workshop on Recommender Systems for Technology Enhanced Learning*, Elsevier's Procedia CS, 2010, pp. 2811-2819.

[18] N. Thai-Nghe, L. Drumond, T. Horváth, A. Krohn-Grimberghe, A. Nanopoulos, and L. Schmidt-Thieme, "Educational Recommender systems and technologies: Practices and challenges in factorization

techniques for predicting student performance," *IGI Global*, 2011, pp. 1-25.
[19] A. Toscher and M. Jahrer, (2010). "Collaborative filtering applied to educational data mining," presented at ACM International Conference on Knowledge Discovery and Data Mining, KDD Cup Workshop. Available: http://www.commendo.at/UserFiles/commendo/File/KDDCup2010_Toescher_Jahrer.pdf.
[20] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, vol. 13, pp. 556-562, 2001.
[21] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323-2326, 2000.
[22] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp.1373- 1396, 2003.
[23] D. Cai, X. He, X. Wang, H. Bao, and J. Han, "Locality preserving nonnegative matrix factorization," in *Proc. The 21st International Joint Conference on Artificial Intelligence*, Pasadena, CA, United States, 2009, pp. 1010-1015.
[24] Y. D. Kim, "Weighted nonnegative matrix factorization," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2009, pp. 1541-1544.
[25] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd Ed. New York: Springer, 2009.
[26] T. Barnes and D. Bitzer, "Fault tolerant teaching and automated knowledge assessment," in *Proc. the ACM Southeast Conference*, Raleigh, NC, 2002, pp. 125-132.
[27] E. Ayers, R. Nugent, and N. Dean, "Skill set profile clustering based on student capability vectors computed from online tutoring data," in *Proc. the 1st International Conference on Education Data Mining* in Montreal Canada, 2008, pp. 218-225.
[28] T. Chen, Z. Zheng, Q. Lu, W. Zhang, and Y. Yu, (2011, July 11). "Feature-based matrix factorization," working paper, Apex Data & Knowledge Management Lab, Arxiv preprint arXiv:1109.2271, Shanghai Jiao Tong University. Available: http://www0.cs.ucl.ac.uk/staff/Weinan.Zhang/papers/fbmf.pdf