MPMA: Mixture Probabilistic Matrix Approximation for Collaborative Filtering

Chao Chen,[1,*] **Dongsheng Li,**[2] **Qin Lv,**[3] **Junchi Yan,**[2,4] **Stephen M. Chu,**[2] **Li Shang**[1,3]
[1]Tongji University, Shanghai, P.R. China, 201804
[2]IBM Research - China, Shanghai, P.R. China, 201203
[3]Univeristy of Colorado Boulder, Boulder, Colorado, USA, 80309
[4]East China Normal University, Shanghai, P.R. China, 200062
chench.resch@gmail.com, {ldsli, jcyan, schu}@cn.ibm.com, {qin.lv, li.shang}@colorado.edu

Abstract

Matrix approximation (MA) is one of the most popular techniques for collaborative filtering (CF). Most existing MA methods train user/item latent factors based on a user-item rating matrix and then use the global latent factors to model all users/items. However, globally optimized latent factors may not reflect the unique interests shared among only subsets of users/items, without which unique interests of users may not be accurately modelled. As a result, existing MA methods, which cannot capture the uniqueness of different user/item, cannot provide optimal recommendation.

In this paper, a mixture probabilistic matrix approximation (MPMA) method is proposed, which unifies globally optimized user/item feature vectors (on the entire rating matrix) and locally optimized user/item feature vectors (on subsets of user/item ratings) to improve recommendation accuracy. More specifically, in MPMA, a method is developed to find both globally and locally optimized user/item feature vectors. Then, a Gaussian mixture model is adopted to combine global predictions and local predictions to produce accurate rating predictions. Experimental study using MovieLens and Netflix datasets demonstrates that MPMA outperforms five state-of-the-art MA based CF methods in recommendation accuracy with good scalability.

1 Introduction

Collaborative filtering (CF) methods have achieved great success in today's recommender systems, among which matrix approximation (MA) is one of the most popular techniques. In MA-based CF methods, both users and items are characterized by vectors of latent factors inferred from the user-item rating matrix, and these latent factors are used to make rating predictions [Adomavicius and Tuzhilin, 2005; Su and Khoshgoftaar, 2009]. Most existing MA-based methods [Srebro *et al.*, 2004; Salakhutdinov and Mnih, 2007; Koren, 2008] rely on globally optimized user/item latent factors to produce recommendations. However, in many real-world applications, if we take the globally optimized latent factors as "common interests", then subsets of users may share "unique interests" that are not covered by the "common interests" [Xu *et al.*, 2012]. Therefore, if MA methods only consider the "common interests" without considering "unique interests", the recommendations may not be optimal for the "unique" subset of each user. As pointed out by Koren et al. [2008], MA models can effectively estimate overall structures that relate simultaneously to most or all items, but they perform poorly at detecting strong associations among a small set of items.

Recently, matrix clustering [Xu *et al.*, 2012; Lee *et al.*, 2013] and community detection [Zhang *et al.*, 2013] methods have been proposed to discover the localized relationships among subsets of users/items. In these methods, local user/item latent factors are trained within clusters/communities, and then recommendations are produced by local models. However, these methods often rely on only local models and fail to incorporate the global latent factors in recommendation, which may compromise recommendation quality due to insufficient data in local clusters or communities [Chen *et al.*, 2015]. In summary, MA-based CF methods only relying on either global latent factors or local latent factors alone compromise recommendation accuracy. Therefore, there is need to design new MA-based CF methods which can incorporate both global latent factors and local latent factors.

In this paper, we developed a new method to model each user/item using local and global latent factors to capture both localized relationships in user-item subgroups and common associations among all users and items. Following multitask feature learning techniques [Evgeniou and Pontil, 2007; Ando and Zhang, 2005], we share global user/item latent factors across user-item subgroups, so that the local latent factors can be trained without suffering from insufficient data issue. In a Bayesian perspective, the proposed mixture probabilistic matrix approximation (MPMA) method assumes every user-item rating can be depicted by a Gaussian mixture model containing three components: (1) a global model that

[*] Chao Chen and Dongsheng Li contributed equally to this work.

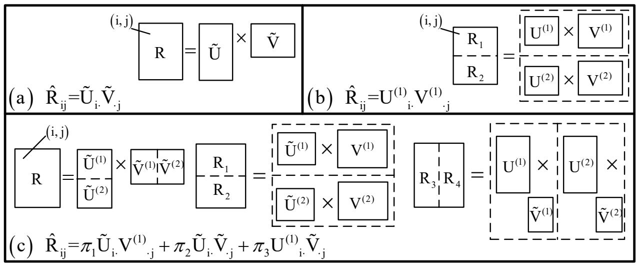

Figure 1: Comparison of three kinds of matrix approximation models for collaborative filtering. (a) standard low-rank model, (b) clustering-based model, and (c) the proposed MPMA model.

captures the global latent factors of all users and items, (2) a user-based local model that captures the local latent factors of the corresponding subset of users, and (3) an item-based local model that captures the local latent factors of the corresponding subset of items. In MPMA, a new optimization problem is first defined and solved to obtain the global/local latent factors for each user/item. A pipeline-based learning method is proposed to train the user/item latent factors in parallel. Finally, recommendation scores are generated by combining the scores from different components of the mixture model. The proposed MPMA method is evaluated using two real-world datasets (MovieLens and Netflix), and the experimental results demonstrate that MPMA achieves better recommendation accuracy than five state-of-the-art MA-based CF methods while achieving good efficiency.

2 Related Work

Recommender systems have become increasingly popular in recent years, aiming to provide personalized recommendations for products that suit users' tastes [Adomavicius and Tuzhilin, 2005]. Among existing recommender solutions, collaborative filtering (CF) is widely used for simplicity of implementation and high quality of recommendation. Early CF algorithms focus on memory-based approaches, such as user-based [Herlocker *et al.*, 1999] and item-based [Sarwar *et al.*, 2001] methods, which make rating predictions based on similarities between users/items. However, these approaches suffer from the data sparsity problem, because the user/item similarities cannot be calculated accurately without sufficient ratings.

Recently, matrix approximation-based CF methods have been proposed to alleviate the data sparsity issue. Billsus et al. [1998] first introduced SVD to the domain of collaborative filtering. Later on, a maximum-margin matrix factorization (MMMF) method was proposed by Srebro et al. [2004]. Salakhutdinov et al. [2007] first proposed a probabilistic matrix factorization (PMF) method, and later constructed BPMF — a Bayesian extension of PMF method [Salakhutdinov and Mnih, 2008]. Koren et al. [2008] pointed out that these models are generally effective at estimating overall structure that relates simultaneously to most or all items, but perform poorly at detecting strong associations among a small set of closely related items. To address this issue, recent work adopted matrix clustering techniques [Xu *et al.*, 2012] and community detection methods [Zhang *et al.*, 2013] to find user/item clusters with strong correlations to improve recommendation accuracy. Mackey et al. [2011] and Lee et al. [2013] followed divide-and-conquer methodology, which divides the MA task into smaller subproblems, solves these subproblems in parallel, and then combines the recommendations of sub-models to achieve better accuracy. However, these methods mainly focus on ratings inside clusters and ignore the majority of user ratings outside clusters. Since training data are often insufficient in the detected clusters, the performance of local models may degrade due to severe overfitting [Chen *et al.*, 2015]. Figure 1 summarizes these two types of MA models in (a) and (b), respectively.

The objective of this work is to unify localized relationships in user-item subgroups and common associations among all users and items to improve the recommendation accuracy. The most related existing works are Collective Matrix Factorization (CMF) [Singh and Gordon, 2008] and Group-Sparse Matrix Factorization (GSMF) [Yuan *et al.*, 2014]. CMF shares parameters among factors when decomposing multiple matrices represented for multiple relations to learn different type of user behaviors. And GSMF uses group sparsity regularization to automatically transfer information among multiple types of behaviors. Different from previous works, the proposed work only utilizes rating matrix without requiring additional information and directly models each user/item by global (common) and local (private) features, and then shares global features across multiple tasks. We illustrate this idea in Figure 1 (c), where we can see, items for users in R_1 are modeled by both global features $\tilde{V}$ and local features $V^{(1)}$, and the global user features $\tilde{U}^{(1)}$ and $\tilde{U}^{(2)}$ are shared across multiple tasks. Notably, the proposed MPMA method can learn multiple related tasks simultaneously, since multi-task feature learning has been empirically and theoretically proved to often significantly improve model performance relative to learning each task independently [Evgeniou and Pontil, 2007; Ando and Zhang, 2005]. For better understanding, we introduce the proposed MPMA method in a Bayesian perspective.

3 Mixture Probabilistic Matrix Approximation

This section first formulates the MPMA problem. The method for learning user/item feature vectors in the mixture model of MPMA is then described. Finally, the rating prediction method based on the mixture model is presented in detail.

3.1 Problem Formulation

Let's first introduce the notations used in this paper. Upper case letters, such as R, U, and V, denote matrices. For matrix $R \in \mathbb{R}^{m\times n}$, we denote $R_{i\cdot}$ as the i-th row vector, $R_{\cdot j}$ as the j-th column vector, and R_{ij} as the entry in the i-th row and j-th column. In addition, the Frobenius norm is adopted in this paper, which is defined as $||R||^2 := \sum_{i=1}^{m}\sum_{j=1}^{n}R_{ij}^2$.

MPMA assumes that each user/item is modeled by two levels of latent factors: (1) global latent factors shared across all users/items, and (2) local latent factors shared

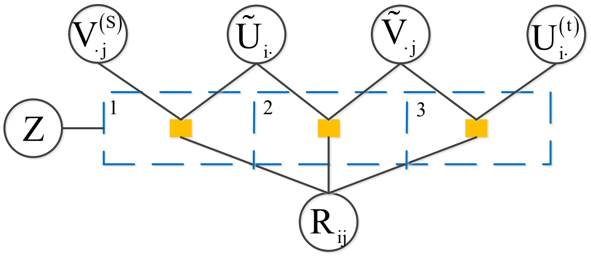

Figure 2: A high-level illustration of the MPMA method. Every rating R_{ij} is described by a mixture model with a set of latent variables Z, which consists of (1) user-based local model, (2) global model, and (3) item-based local model.

within subsets of users/items. Furthermore, as shown in Figure 2, we introduce a new mixture feature vector model. In this model, each user i (item j) is characterized by a global feature vector $\tilde{U}_{i\cdot}(\tilde{V}_{\cdot j})$ and a local feature vector $U^{(t)}_{i\cdot}(V^{(s)}_{\cdot j})$. Meanwhile, each rating R_{ij} is also described by the mixture model with latent variables Z, which consists of local user model $\mathcal{N}(R_{ij}|\tilde{U}_{i\cdot}V^{(s)}_{\cdot j}, \sigma_1^2\mathbf{I})$ (left), local item model $\mathcal{N}(R_{ij}|U^{(t)}_{i\cdot}\tilde{V}_{\cdot j}, \sigma_3^2\mathbf{I})$ (right), and global model $\mathcal{N}(R_{ij}|\tilde{U}_{i\cdot}\tilde{V}_{\cdot j}, \sigma_2^2\mathbf{I})$ (middle).

To discover user-item subgroups sharing similar unique interests, many existing clustering techniques or community detection methods can be adopted, such as K-means++ method [Arthur and Vassilvitskii, 2007], and Bregman Co-clustering method [Banerjee *et al.*, 2007], etc. Without losing any generality, we assume that users are divided into g groups, and the rating matrix of the t-th user group is denoted as $R^{(t)}_U (t \in [1, g])$. Similarly, items are divided into f groups, and the rating matrix of the s-th item group is denoted as $R^{(s)}_V (s \in [1, f])$. For each user-item pair (i, j), we assume user i is in the t-th user group and item j is in the s-th item group. Then, the conditional distributions of the three corresponding components (as illustrated in Figure 2) can be defined by three Gaussian models, as follows:

$$\begin{aligned}
p(R_{ij}|\tilde{U}_{i\cdot}, V^{(s)}_{\cdot j}, \sigma_1^2) &= \mathcal{N}(R_{ij}|\tilde{U}_{i\cdot}V^{(s)}_{\cdot j}, \sigma_1^2) \qquad (1)\\
p(R_{ij}|\tilde{U}_{i\cdot}, \tilde{V}_{\cdot j}, \sigma_2^2) &= \mathcal{N}(R_{ij}|\tilde{U}_{i\cdot}\tilde{V}_{\cdot j}, \sigma_2^2) \qquad (2)\\
p(R_{ij}|U^{(t)}_{i\cdot}, \tilde{V}_{\cdot j}, \sigma_3^2) &= \mathcal{N}(R_{ij}|U^{(t)}_{i\cdot}\tilde{V}_{\cdot j}, \sigma_3^2) \qquad (3)
\end{aligned}$$

where $\tilde{U}_{i\cdot}$ $(\tilde{V}_{\cdot j})$ is the global feature vector of the i-th user (j-th item), and similarly, $U^{(t)}_{i\cdot}$ $(V^{(s)}_{\cdot j})$ is the local feature vector of the i-th user (j-th item) for the t-th user group (s-th item group). Following the idea of probabilistic matrix factorization (PMF) [Salakhutdinov and Mnih, 2007], zero-mean spherical Gaussian priors are placed on all user and item feature vectors. Thus, each global/local feature matrix can be modeled as follows:

$$\begin{aligned}
p(\tilde{U}|\sigma^2_{\tilde{U}}\mathbf{I}) &= \prod_{i=1}^{m}\mathcal{N}(\tilde{U}_{i\cdot}|0, \sigma^2_{\tilde{U}}\mathbf{I}) \qquad (4)\\
p(\tilde{V}|\sigma^2_{\tilde{V}}\mathbf{I}) &= \prod_{j=1}^{n}\mathcal{N}(\tilde{V}_{\cdot j}|0, \sigma^2_{\tilde{V}}\mathbf{I}) \qquad (5)\\
p(U^{(t)}|\sigma^2_{U^{(t)}}\mathbf{I}) &= \prod_{i=1}^{m}\mathcal{N}(U^{(t)}_{i\cdot}|0, \sigma^2_{U^{(t)}}\mathbf{I}) \qquad (6)\\
p(V^{(s)}|\sigma^2_{V^{(s)}}\mathbf{I}) &= \prod_{j=1}^{n}\mathcal{N}(V^{(s)}_{\cdot j}|0, \sigma^2_{V^{(s)}}\mathbf{I}) \qquad (7)
\end{aligned}$$

where $\sigma^2_\cdot$ denotes the co-variance matrix of each random variable and $\mathbf{I}$ denotes the identity matrix. Then, the log of the posterior distribution over the user and item features is given by

$$\begin{aligned}
&\ln p(\tilde{U}, \tilde{V}, U^{(1)}, \ldots, U^{(g)}, V^{(1)}, \ldots, V^{(f)}|R)\\
&= \sum_{s=1}^{f}\sum_{t=1}^{g}\sum_{\rho(i)=s}\sum_{\rho(j)=t}\ln\Big\{\mathrm{I}_{ij}\Big(\pi_1 p(R_{ij}, \tilde{U}, V^{(s)}) \qquad (8)\\
&+ \pi_2 p(R_{ij}, \tilde{U}, \tilde{V}) + \ \pi_3 p(R_{ij}, U^{(t)}, \tilde{V})\Big)\Big\} + C,
\end{aligned}$$

where C is a constant that does not depend on the parameters. Unfortunately, it is very difficult to directly maximize the log-posterior in Equation (8). Therefore, we try to find an approximate solution via maximizing the lower bound of Equation (8). Based on Jensen's inequality, the lower bound of Equation (8) can be obtained as follows:

$$\begin{aligned}
&\ln p(\tilde{U}, \tilde{V}, U^{(1)}, \ldots, U^{(g)}, V^{(1)}, \ldots, V^{(f)}|R)\\
&\geq C_1 + \sum_{s=1}^{f}\sum_{t=1}^{g}\sum_{\rho(i)=s}\sum_{\rho(j)=t}\mathrm{I}_{ij}\Big\{\pi_2\ln p(R_{ij}, \tilde{U}, \tilde{V}) \qquad (9)\\
&+ \pi_1\ln p(R_{ij}, \tilde{U}, V^{(s)}) + \ \pi_3\ln p(R_{ij}, U^{(t)}, \tilde{V})\Big\}
\end{aligned}$$

where I_{ij} is an indicator function. $\mathrm{I}_{ij} = 1$ if user i rates item j in the training data, and $\mathrm{I}_{ij} = 0$ otherwise. Then, applying Equations (1) to (7) in Equation (9), the optimization objective becomes:

$$\begin{aligned}
\mathcal{L}' = C_2 &- \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{\pi_2}{2\sigma_2^2}\mathrm{I}_{ij}(R_{ij} - \tilde{U}_{i\cdot}\tilde{V}_{\cdot j})^2\\
&- \sum_{s=1}^{f}\sum_{t=1}^{g}\sum_{\rho(i)=s}\sum_{\rho(j)=t}\frac{\pi_1}{2\sigma_1^2}\mathrm{I}_{ij}(R_{ij} - \tilde{U}_{i\cdot}V^{(s)}_{\cdot j})^2\\
&- \sum_{s=1}^{f}\sum_{t=1}^{g}\sum_{\rho(i)=s}\sum_{\rho(j)=t}\frac{\pi_3}{2\sigma_2^2}\mathrm{I}_{ij}(R_{ij} - U^{(t)}_{i\cdot}\tilde{V}_{\cdot j})^2 \qquad (10)\\
&- \sum_{s=1}^{g}\sum_{i=1}^{m}\frac{\pi_3}{2\sigma^2_{U^{(t)}}}U^{(t)}_{i\cdot}[U^{(t)}_{i\cdot}]' - \frac{\pi_1 + \pi_2}{2\sigma^2_{\tilde{U}}}\sum_{i=1}^{m}\tilde{U}_{i\cdot}\tilde{U}'_{i\cdot}\\
&- \sum_{t=1}^{f}\sum_{j=1}^{n}\frac{\pi_1}{2\sigma^2_{V^{(s)}}}[V^{(s)}_{\cdot j}]'V^{(s)}_{\cdot j} - \frac{\pi_2 + \pi_3}{\sigma^2_{\tilde{V}}}\sum_{j=1}^{n}\tilde{V}'_{\cdot j}\tilde{V}_{\cdot j}
\end{aligned}$$

To maximize the objective function defined in (10), it is equivalent to minimize the sum of squared error loss function with quadratic regularization terms as follows:

$$
\begin{aligned}
&\min_{\tilde{U},\tilde{V},U^{(1)},\ldots,U^{(g)},V^{(1)},\ldots,V^{(f)}} \mathcal{L}(\tilde{U},\tilde{V},U^{(1)},\ldots,V^{(f)}) \\
&=\| I\otimes(R-\tilde{U}\tilde{V}) \|^2 +\lambda_1 \| \tilde{U} \|^2 +\lambda_2 \| \tilde{V} \|^2 \qquad (11)\\
&+\sum\nolimits_{s\in[f]} \alpha_s \| I_U^{(s)} \otimes (R_U^{(s)} - \tilde{U}^{(s)}V^{(s)}) \|^2 \qquad (12)\\
&+\sum\nolimits_{t\in[g]} \beta_t \| I_V^{(t)} \otimes (R_V^{(t)} - U^{(t)}\tilde{V}^{(t)}) \|^2 \qquad (13)\\
&+\sum\nolimits_{s\in[f]} \lambda_3 \| V^{(s)} \|^2 +\sum\nolimits_{t\in[g]} \lambda_4 \| U^{(t)} \|^2 \qquad (14)
\end{aligned}
$$

where $\alpha_s = (\pi_1\sigma^2)/(\pi_2\sigma^2_{(s)})$ and $\beta_t = (\pi_3\sigma^2)/(\pi_2\sigma^2_{(t)})$ are the weights for local models by giving the weight of global model to 1. Moreover, $\lambda_1 = [(\pi_1+\pi_2)\sigma_2^2]/(\pi_2\sigma^2_{\tilde{U}})$, $\lambda_2 = [(\pi_2+\pi_3)\sigma_2^2]/(\pi_2\sigma^2_{\tilde{V}})$, $\lambda_3 = (\pi_1\sigma_2^2)/(\pi_2\sigma^2_{\tilde{V}})$ and $\lambda_4 = (\pi_3\sigma_2^2)/(\pi_2\sigma^2_{\tilde{U}})$ are the regularization parameters.

Based on Equations (11) to (14), the key characteristics of MPMA are summarized as follows:

- The terms in Equation (11) are the same as the optimization objective of standard SVD method, which ensure that the learned $\tilde{U}$ and $\tilde{V}$ can accurately estimate the overall structure and avoid overfitting in user/item feature vectors.

- The terms in Equation (12) and Equation (13) can help learn local latent factors based on the global latent factors $\tilde{U}$ and $\tilde{V}$, and parameters α_s and β_t can be employed to control the contribution of individual user/item clusters. Meanwhile, the global latent factors will be updated during the optimization of local latent factors, so that the global latent factors of MPMA will be different from those in standard SVD method.

- The standard SVD method can be viewed as a special case of MPMA by setting all $\alpha_s = \beta_t = 0$. More specifically, we refer to the model with only local user features as u-MPMA, the model with only local item features as i-MPMA, and the one with both of them as MPMA. Empirical study shows both u-MPMA and i-MPMA can improve recommendation accuracy and MPMA achieves better improvement of recommendation accuracy than u-MPMA and i-MPMA.

3.2 Optimization Algorithm

After defining the optimization objective, we present how to solve the optimization problem in this section. As shown by Srebro et al. [2003], the problem defined in Equations (11) to (14) is a difficult non-convex optimization problem, because the introduced weights lead to significant changes in the critical point structure. To tackle this problem, we develop a stochastic gradient descent (SGD) based method.

In SGD, we first compute the partial derivatives of parameters, and then iteratively update the parameters until convergence. The partial derivatives of the loss function with respect to global features $\tilde{U}$ and $\tilde{V}$ can be computed as follows:

$$
\begin{aligned}
\frac{\partial\mathcal{L}}{\partial\tilde{U}^{(s)}} &= \lambda_1\tilde{U}^{(s)} + I_U^{(s)}\otimes(\tilde{U}^{(s)}\tilde{V} - R_U^{(s)})\tilde{V}' \\
&+ \alpha_s I_U^{(s)}\otimes(\tilde{U}^{(s)}V^{(s)} - R_U^{(s)})[V^{(s)}]' \qquad (15)\\
\frac{\partial\mathcal{L}}{\partial\tilde{V}^{(t)}} &= \lambda_2\tilde{V}^{(t)} + I_V^{(t)}\otimes(\tilde{U}\tilde{V}^{(t)} - R_V^{(t)})'\tilde{U} \\
&+ \beta_t I_V^{(t)}\otimes(U^{(t)}\tilde{V}^{(t)} - R_V^{(t)})'U^{(t)} \qquad (16)
\end{aligned}
$$

The partial derivatives of local feature vectors $U^{(t)}$ and $V^{(s)}$ are given by

$$
\begin{aligned}
\frac{\partial\mathcal{L}}{\partial U^{(t)}} &= \lambda_3 U^{(t)} + \beta_t I_V^{(t)}\otimes(U^{(t)}\tilde{V}^{(t)} - R_V^{(t)})[\tilde{V}^{(t)}]' (17)\\
\frac{\partial\mathcal{L}}{\partial V^{(s)}} &= \lambda_4 V^{(s)} + \alpha_s I_U^{(s)}\otimes(\tilde{U}^{(s)}V^{(s)} - R_U^{(s)})'\tilde{U}^{(s)} (18)
\end{aligned}
$$

Compared with standard matrix approximation based methods, the proposed method requires more computation time due to the extra local features. However, we can see that the parameter update stream consists of two processing stages: S_1) updating global feature vectors and S_2) updating local feature vectors. If the S_1 stage is performed on a submatrix which has no overlapping rows and columns with the submatrix that S_2 stage is performed, then we can construct a pipeline to perform S_1 and S_2 in parallel.

The detailed pipeline-based learning method is presented in Algorithm 1. In line 7 and 8, S_1 and S_2 can run in parallel, so that the overall training time of MPMA can be reduced. Note that following the training data scheduling method in [Recht and Ré, 2013], we can set up more computing threads in Algorithm 1 to further improve the training efficiency.

Algorithm 1 Efficient Pipeline-based Learning Algorithm

```
Input: Rating matrices $R$ for all users and items, $R_U^{(s)}(s \in [1,g])$ for each user group, and $R_V^{(t)}(t \in [1,f])$ for each item group. $\alpha_s, \beta_t, \lambda$ are parameters in the mixture model, and $r$ is the rank for MA.
Output: $(\tilde{U},\tilde{V},U^{(1)},\ldots,U^{(g)},V^{(1)},\ldots,V^{(f)})$
1: //building model
2: Randomly initialize $\tilde{U} \in \mathbb{R}^{m\times r}$, $U^{(s)} \in \mathbb{R}^{m\times r}(s \in [1,g])$, $\tilde{V} \in \mathbb{R}^{r\times n}$ and $V^{(t)} \in \mathbb{R}^{r\times n}(t \in [1,f])$;
3: while not converged do
4:     Choose $i \neq i' \in [1,g]$ and $j \neq j' \in [1,f]$;
5:     $R^{(i,j)} = R_U^{(i)} \cap R_V^{(j)}$, $R^{(i',j')} = R_U^{(i')} \cap R_V^{(j')}$;
6:     //Update in parallel:
7:     S1: update $\tilde{U}$ and $\tilde{V}$ w.r.t. $R^{(i',j')}$ by Eq. (15)(16);
8:     S2: update $U^{(i)}$ and $V^{(j)}$ w.r.t. $R^{(i,j)}$ by Eq. (17)(18);
9: end while
```

3.3 Rating Prediction

In MPMA, each user-item rating is characterized by a mixture model, so that we can estimate the "weight" of each component in the mixture model — π_k and then apply the mixture

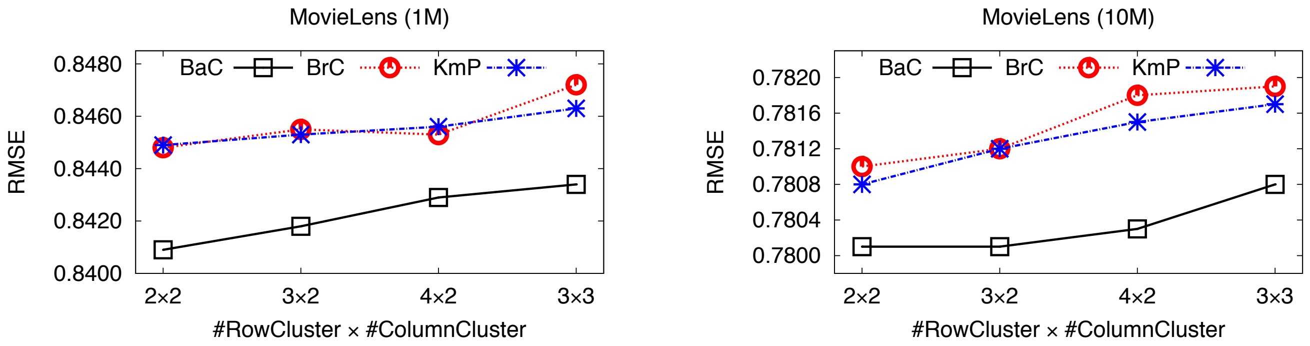

Figure 3: Performance of applying different clustering methods in MPMA with rank = 20, while the number of row and column clusters varies in $\{2 \times 2, 3 \times 2, 4 \times 2, 3 \times 3\}$ on MovieLens 1M (left) and MovieLens 10M (right).

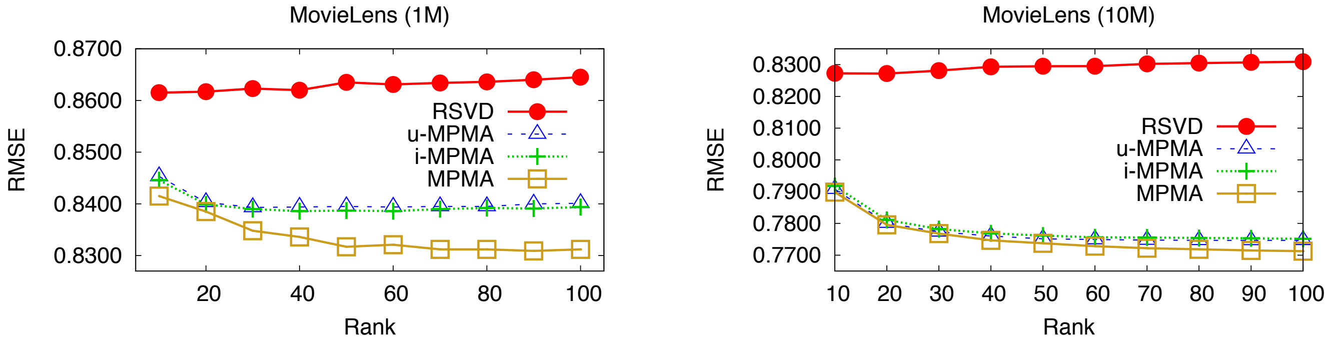

Figure 4: Performance of local cluster-based constraints with different rank r ranging in $[10, 100]$ on MovieLens 1M (left) and MovieLens 10M (right).

components to predict user-item ratings. The hidden parameters π_k ($k \in [1, 3]$) can be obtained with the EM method as follows:

E-step:

$$\gamma(Z_{ij}^k) = \frac{\pi_k \mathcal{N}(R_{ij}|F_k, \sigma_k^2)}{\sum_{k' \in [1,3]} \pi_{k'} \mathcal{N}(R_{ij}|F_{k'}, \sigma_k^2)} \tag{19}$$

$$F_1 = \{\tilde{U}_{i\cdot}, V_{\cdot j}\}, F_2 = \{\tilde{U}_{i\cdot}, \tilde{V}_{\cdot j}\}, F_3 = \{U_{i\cdot}, \tilde{V}_{\cdot j}\} \tag{20}$$

M-step:

$$\sigma_k^2 = \frac{1}{N_k} \sum_{ij} \gamma(Z_{ij}^k)(R_{ij} - R_{ij}^{(k)})^2, \pi_k = \frac{N_k}{N} \tag{21}$$

$$N_k = \sum_{ij} \gamma(Z_{ij}^k), N = \sum_k N_k, \tag{22}$$

$$R_{ij}^{(1)} = \tilde{U}_{i\cdot} V_{\cdot j}^{(s)}, R_{ij}^{(2)} = \tilde{U}_{i\cdot} \tilde{V}_{\cdot j}, R_{ij}^{(3)} = U_{i\cdot}^{(t)} \tilde{V}_{\cdot j} \tag{23}$$

where F_k denotes the user and item features for the k-th component in the mixture model. After obtaining π_k, we can use the mean of mixture model to estimate the missing rating of user i on item j as follows:

$$\hat{R}_{ij} = \pi_1 \tilde{U}_{i\cdot} V_{\cdot j}^{(s)} + \pi_2 \tilde{U}_{i\cdot} \tilde{V}_{\cdot j} + \pi_3 U_{i\cdot}^{(t)} \tilde{V}_{\cdot j} \tag{24}$$

where user i (item j) belongs to the t-th user group (s-th item group).

4 Experimental Results

This section evaluates the proposed MPMA method on three well-known datasets which have been widely used for evaluating recommendation algorithms – 1) MovieLens 1M ($\sim 10^6$ ratings of $6,040$ users on $3,706$ items), 2) MovieLens 10M ($\sim 10^7$ ratings of $69,878$ users on $10,677$ items), and 3) Netflix ($\sim 10^8$ ratings of $480,189$ users on $17,770$ items). The root mean square error (RMSE) is adopted as the evaluation metric for recommendation accuracy, which can be computed as $\sqrt{\frac{1}{|T|} \sum_{(u,i) \in T} (\hat{\mathrm{R}}_{ui} - \mathrm{R}_{ui})^2}$, where T denotes the set of ratings in test data and $|T|$ is the number of ratings in test data.

4.1 Sensitivity Analysis

In this study, we evaluate how the recommendation accuracy of MPMA varies with different parameter settings on MovieLens (1M) and MovieLens (10M) datasets, i.e., the number of clusters and rank.

Accuracy vs. Clustering methods

Figure 3 analyzes the impact of different clustering methods with different numbers of row and column clusters f and g on MovieLens (1M) and MovieLens (10M) respectively, where we use three commonly-used and typical clustering method: K-means++(KmP) method [Arthur and Vassilvitskii, 2007], Balanced clustering(BaC) method [Banerjee and Ghosh, 2002], and Bregman Co-clustering(BrC) method [Banerjee *et al.*, 2007]. Note that, MPMA method is orthogonal to clustering algorithms, so other clustering methods can also be adopted in MPMA. In Figure 3, the cases of 1×2 and 2×1 clusterings are not considered, because those two cases are i-MPMA and u-MPMA, respectively, and the results will be presented in Figure 4.

We can see from the results that MPMA with BaC method can outperform MPMA with both KmP and BrC methods in recommendation accuracy in all cases. This is because both of KmP and BrC methods will often produce one or a few large sparse clusters, which has been proved to be harder to provide personal recommendations [Su and Khoshgoftaar, 2009]. In addition, we can see that the accuracies of all methods decrease as f and g increase. This is due to the fact that each co-cluster contains less user-item ratings as the number of user/item clusters increases, resulting in insufficient data for model training.

Accuracy vs. Latent factors

As shown in Figure 4, the proposed u-MPMA (with local user features) and i-MPMA (with local item features) methods can both outperform the classic RSVD method on all ranks. This indicates that both local user features and local item features can help improve the recommendation accuracy. Moreover, the proposed MPMA method, which adopts both local user features and local item features, outperforms all other three methods (RSVD, u-MPMA, and i-MPMA) with all ranks. This indicates that the benefits of local user features and local item features are orthogonal, and thus should be both adopted in collaborative filtering.

Accuracy vs. Rank

Figure 4 also studies the effect of local user/item features with rank r ranging in $[10, 100]$ on both the MovieLens (1M) and MovieLens (10M) datasets. As shown in the results, higher ranks will lead to better accuracy when r increases from 10 to 100, and the results get stable when r is larger than 50. But the recommendation accuracy of RSVD decreases slightly as r increases, which means large ranks will cause overfitting on RSVD. But in the proposed method, overfitting is not observed even when rank r is as large as 100, which indicates that the introduced local features can increase robustness of the proposed method.

Efficiency vs. Rank

Figure 5 compares the computation efficiency of RSVD method and MPMA method with different ranks. We can see from the results that MPMA achieves comparable efficiency with RSVD with the help of the pipeline-based learning framework (Algorithm 1), although MPMA has to learn many more local features. Note that, more pipelines can be set to further enhance efficiency by dividing more user/item groups. This demonstrates that MPMA can achieve good scalability on large datasets.

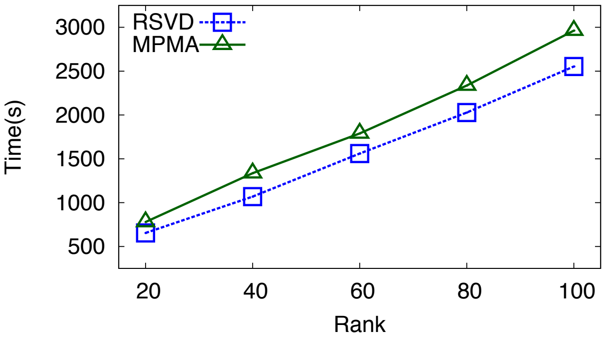

Figure 5: Efficiency comparison of RSVD and MPMA with different ranks r on the MovieLens (10M) dataset.

Table 1: RMSEs of the proposed MPMA method and the other five state-of-the-art methods — NMF [Lee and Seung, 2001], RSVD [Paterek, 2007], BPMF [Salakhutdinov and Mnih, 2008], APG [Toh and Yun, 2010], GSMF [Yuan *et al.*, 2014].

	MovieLens (10M)	Netflix
NMF	0.8832 ± 0.0007	0.9396 ± 0.0002
RSVD	0.8271 ± 0.0009	0.8534 ± 0.0001
BPMF	0.8195 ± 0.0006	0.8420 ± 0.0003
APG	0.8098 ± 0.0005	0.8476 ± 0.0028
GSMF	0.8012 ± 0.0011	0.8420 ± 0.0006
MPMA	**0.7712** ± **0.0002**	**0.8139** ± **0.0003**

4.2 Performance Comparison

In this study, we compare the recommendation accuracy of the proposed method against five state-of-the-art matrix approximation based CF methods: NMF [Lee and Seung, 2001], RSVD [Paterek, 2007], BPMF [Salakhutdinov and Mnih, 2008], APG [Toh and Yun, 2010] and GSMF [Yuan *et al.*, 2014]. We emphasize the comparison between MPMA and GSMF, because GSMF is the latest work related to the proposed MPMA method and empirically proves to be better than CMF [Singh and Gordon, 2008].

In the following experiments, all results are presented by averaging the results over five different random train/test splits with the ratio of 9:1. For the proposed method, we set rank $r = 100$ and $f \times g$ as 2×2, learning rate $v = 0.001$. For the parameters from Equation 11 to 14, we set all α and β values to 1 due to the hardness of estimating all σ values, and set all λ values to 0.06 for L_2-regularization. Optimal parameters of the compared methods are chosen from their papers.

As we can see from Table 1, MPMA significantly outperforms all the other five matrix approximation based CF methods on both datasets. Compared with NMP, RSVD, BPMF, APG and GSMF, the main reasons that the proposed MPMA method can make more accurate recommendations are 1) local latent factors are adopted by MPMA to better understand user interests and 2) a mixture model is adopted to combine both local predictions and global predictions to achieve more accurate predictions.

5 Conclusion

Existing matrix approximation based collaborative filtering methods rely on either global latent factors or local latent factors of users and items, and cannot provide optimal user/item modelling and the most accurate recommendations. In this paper, a mixture probabilistic matrix approximation (MPMA) method is proposed, which unifies global latent factors and local latent factors of users and items by a Gaussian mixture model to improve recommendation accuracy. Experimental study on real-world datasets demonstrates that the proposed MPMA method can outperform five state-of-the-art MA-based collaborative filtering methods in recommendation accuracy. Furthermore, the proposed MPMA method can be trained by a pipeline-based method, so that it is scalable on large datasets.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61233016, and the National Science Foundation of USA under Grant Nos. 0954157, 1251257, 1334351, and 1442971.

References

[Adomavicius and Tuzhilin, 2005] Gediminas Adomavicius and Alexander 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 Engeering*, 17(6):734–749, 2005.

[Ando and Zhang, 2005] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *The Journal of Machine Learning Research*, 6:1817–1853, 2005.

[Arthur and Vassilvitskii, 2007] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *SODA '07*, pages 1027–1035, 2007.

[Banerjee and Ghosh, 2002] Arindam Banerjee and Joydeep Ghosh. On scaling up balanced clustering algorithms. In *SDM '02*, pages 333–349, 2002.

[Banerjee *et al.*, 2007] Arindam Banerjee, Inderjit Dhillon, Joydeep Ghosh, Srujana Merugu, and Dharmendra S. Modha. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. *The Journal of Machine Learning Research*, 8:1919–1986, 2007.

[Billsus and Pazzani, 1998] Daniel Billsus and Michael J Pazzani. Learning collaborative information filters. In *ICML '98*, pages 46–54, 1998.

[Chen *et al.*, 2015] Chao Chen, Dongsheng Li, Yingying Zhao, Qin Lv, and Li Shang. WEMAREC: Accurate and scalable recommendation through weighted and ensemble matrix approximation. In *SIGIR '15*, pages 303–312. ACM, 2015.

[Evgeniou and Pontil, 2007] A Evgeniou and Massimiliano Pontil. Multi-task feature learning. *NIPS '07*, 19:41, 2007.

[Herlocker *et al.*, 1999] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR '99*, pages 230–237, 1999.

[Koren, 2008] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD '08*, pages 426–434. ACM, 2008.

[Lee and Seung, 2001] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *NIPS '01*, pages 556–562, 2001.

[Lee *et al.*, 2013] Joonseok Lee, Seungyeon Kim, Guy Lebanon, and Yoram Singer. Local low-rank matrix approximation. In *ICML '13*, pages 82–90, 2013.

[Mackey *et al.*, 2011] Lester W Mackey, Michael I Jordan, and Ameet Talwalkar. Divide-and-conquer matrix factorization. In *NIPS '11*, pages 1134–1142, 2011.

[Paterek, 2007] Arkadiusz Paterek. Improving regularized singular value decomposition for collaborative filtering. In *KDD CUP '07*, volume 2007, pages 5–8, 2007.

[Recht and Ré, 2013] Benjamin Recht and Christopher Ré. Parallel stochastic gradient algorithms for large-scale matrix completion. *Mathematical Programming Computation*, 5(2):201–226, 2013.

[Salakhutdinov and Mnih, 2007] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *NIPS '07*, pages 1257–1264, 2007.

[Salakhutdinov and Mnih, 2008] Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *ICML '08*, pages 880–887. ACM, 2008.

[Sarwar *et al.*, 2001] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW '01*, pages 285–295. ACM, 2001.

[Singh and Gordon, 2008] Ajit P Singh and Geoffrey J Gordon. Relational learning via collective matrix factorization. In *KDD '08*, pages 650–658. ACM, 2008.

[Srebro and Jaakkola, 2003] Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *ICML '03*, pages 720–727, 2003.

[Srebro *et al.*, 2004] Nathan Srebro, Jason Rennie, and Tommi S Jaakkola. Maximum-margin matrix factorization. In *NIPS '04*, pages 1329–1336, 2004.

[Su and Khoshgoftaar, 2009] Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, January 2009. Article ID 421425.

[Toh and Yun, 2010] Kim-Chuan Toh and Sangwoon Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization*, 6(615-640):15, 2010.

[Xu *et al.*, 2012] Bin Xu, Jiajun Bu, Chun Chen, and Deng Cai. An exploration of improving collaborative recommender systems via user-item subgroups. In *WWW '12*, pages 21–30. ACM, 2012.

[Yuan *et al.*, 2014] Ting Yuan, Jian Cheng, Xi Zhang, Shuang Qiu, and Hanqing Lu. Recommendation by mining multiple user behaviors with group sparsity. In *AAAI '14*, 2014.

[Zhang *et al.*, 2013] Yongfeng Zhang, Min Zhang, Yiqun Liu, and Shaoping Ma. Improve collaborative filtering through bordered block diagonal form matrices. In *SIGIR '13*, pages 313–322. ACM, 2013.