Bayesian Probabilistic Multi-Topic Matrix Factorization for Rating Prediction

Keqiang Wang[1], **Wayne Xin Zhao**[2*], **Hongwei Peng**[1], **Xiaoling Wang**[1]
[1]Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, China
[2]School of Information, Renmin University of China, China
{sei.wkq2008, batmanfly}@gmail.com, penghongwei_phw@163.com, xlwang@sei.ecnu.edu.cn

Abstract

Recently, *Local Matrix Factorization* (LMF) [Lee *et al.*, 2013] has been shown to be more effective than traditional matrix factorization for rating prediction. The core idea for LMF is to first partition the original matrix into several smaller submatrices, further exploit local structures of submatrices for better low-rank approximation. Various clustering-based methods with heuristic extensions have been proposed for LMF in the literature. To develop a more principled solution for LMF, this paper presents a Bayesian Probabilistic Multi-Topic Matrix Factorization model. We treat the set of the rated items by a user as a document, and employ latent topic models to cluster items as topics. Subsequently, a user has a distribution over the set of topics. We further set topic-specific latent vectors for both users and items. The final prediction is obtained by an ensemble of the results from the corresponding topic-specific latent vectors in each topic. Using a multi-topic latent representation, our model is more powerful to reflect the complex characteristics for users and items in rating prediction, and enhance the model interpretability. Extensive experiments on large real-world datasets demonstrate the effectiveness of the proposed model.

1 Introduction

Nowadays, recommender systems have played a more and more important role in e-commerce services. A typical task for personalized recommendation is rating prediction which predicts the rating of a user on a given item based on her historical data. Various methods have been proposed in the literature of rating prediction, especially the matrix factorization technique (MF) [Koren *et al.*, 2009]. MF projects users and items into a latent low-dimensional space. Further, the missing entries in the original matrix can be recovered using the dot product between user and item latent vectors. MF has been shown to perform well in many real systems and competitions, such as *Netflix Prize* and *KDD Cup 2011 Recommending Music Items*.

*Corresponding author

Recently, *local matrix factorization* [Lee *et al.*, 2013] has been shown to be more effective than the traditional MF. The original matrix is divided into several smaller submatrices, in which we can exploit local structures for better low-rank approximation. In each submatrix, the standard MF technique is applied to generate submatrix-specific latent vectors for both users and items. Typically, these submatrices are obtained using cluster techniques. By combining the results from multiple local MFs, the original matrix $\mathbf{R}$ is reconstructed by a set of K low-rank submatrices $\{\mathbf{R}^{(1)}, \mathbf{R}^{(2)}, ..., \mathbf{R}^{(K)}\}$ with the corresponding weight matrices $\{\mathbf{L}^{(1)}, \mathbf{L}^{(2)}, ..., \mathbf{L}^{(K)}\}$:

$$\hat{\mathbf{R}}_{um} = \frac{1}{\mathbf{Z}_{um}} \sum_{k=1}^{K} \mathbf{L}_{um}^{(k)} \mathbf{R}_{um}^{(k)} \quad (1)$$

where $\mathbf{Z}_{um} = \sum_{k=1}^{K} \mathbf{L}_{um}^{(k)}$ is the normalizer and $\mathbf{L}_{um}^{(k)}$ indicates the weight for the entry $\mathbf{R}_{um}^{(k)}$ in the submatrix $\mathbf{R}^{(k)}$. Two key issues of such a submatrix-ensemble method are (1) how to generate the submatices and (2) how to set the ensemble weights for submatrices. Several attempts have been made to address these two points, using random sampling [Mackey *et al.*, 2011], extending anchor points with nearest neighbors [Lee *et al.*, 2013; 2014] or co-clustering based matrix partition [Chen *et al.*, 2015].

Although these studies have improved over tradition MF to some extent, there is lack of a more principled approach to characterize the local matrix factorization. By reviewing previous studies [Mackey *et al.*, 2011; Lee *et al.*, 2013; 2014; Chen *et al.*, 2015], we have two important observations: (1) Each submatrix can be considered as a local cluster of users and items; (2) A user or an item has multiple latent representations in different submatrices. Inspired by these two observations, we propose a novel Bayesian Probabilistic Multi-Topic Matrix Factorization model (BPMTMF) for rating prediction. Our model consists of two parts, namely modeling the rated items and modeling the ratings. For the first part, we treat the set of the rated items by a user as a document, and latent topic models are employed to "cluster" items as *topics*, which is a multinomial distribution over the set of items. Subsequently, a user has a distribution (*i.e.,* topical distribution) over the set of topics. Based on such topics, we further set topic-specific latent vectors for both users and items. We integrate the above two parts in a full Bayesian approach. The final prediction by our model is an ensemble of the results

generated from the topic-specific latent vectors in each topic. Our model characterizes the core ideas in Eq. 1 in a Bayesian probabilistic way: each topic can be considered as a cluster. Using a multi-topic latent representation, our model is more powerful to reflect the complex characteristics for users and items in rating prediction. Another important merit of using topics is that our approach has a better model interpretability. Since topic models are effective to discover coherent topical semantics [Blei *et al.*, 2003], the derived topics in our model also group the items that are highly correlated together. In this way, a topic will be more coherent than what have been obtained in previous studies [Mackey *et al.*, 2011; Lee *et al.*, 2013; 2014; Chen *et al.*, 2015]. With topics as contextual information, we can analyze how the rating preference of a user varies in different topical contexts.

Our work presents a Bayesian formulation of local matrix factorization for the first time, which elegantly combines topic models with probabilistic matrix factorization models. By using topics as clusters, our approach has a better model interpretability. Extensive experiments on large real-world datasets demonstrate the effectiveness of the proposed model compared with several competitive baselines.

2 Related Work

In this section, we review the related work.

Matrix Factorization. MF [Paterek, 2007; Mnih and Salakhutdinov, 2007; Koren *et al.*, 2009] is an important kind of model-based collaborative-filtering methods. MF constructs low-rank approximation by projecting users and items into a latent low-dimensional space. Further, Probabilistic Matrix Factorization (PMF) has been proposed by using the Gaussian distribution to model observed ratings with zero-mean spherical Gaussian priors. In essence, PMF can be considered as a probabilistic realization of Regularized Singular Value Decomposition. Further, Salakhutdinov and Mnih [Salakhutdinov and Mnih, 2008] presented a full Bayesian formulation of PMF. As the extensions of MF, biased MF and SVD++ have been proposed in [Koren *et al.*, 2009]. Biased MF incorporates both user bias and item bias, while SVD++ uses implicit feedback to improve user preference modeling.

Local Matrix Factorization. Recently, several studies focus on using the ensemble of submatrices for better low-rank approximation, including DFC [Mackey *et al.*, 2011], LLORMA [Lee *et al.*, 2013; 2014], ACCAMS [Beutel *et al.*, 2015] and WEMAREC [Chen *et al.*, 2015]. These methods partition the original matrix into several smaller submatrices, and a local MF is applied to each submatrix individually. The final predictions are obtained using the ensemble of multiple local MFs. Typically, clustering-based techniques with heuristic adaptations are used for submatrix generation. We give a brief review of these studies. Mackey et al. [Mackey *et al.*, 2011] introduces a Divide-Factor-Combine (DFC) framework, in which the expensive task of matrix factorization is randomly divided into smaller subproblems. LLORMA [Lee *et al.*, 2013; 2014] uses a non-parametric kernel smoothing method to search nearest neighbors; WEMAREC [Chen *et al.*, 2015] employs Bregman co-clustering [Dhillon *et al.*, 2003] techniques to partition the original matrix; ACCAMS adopts an additive co-clustering approach [Shan and Banerjee, 2008] to derive sub-matrices and predict the ratings using a Gaussian distribution. Our work is highly built on the above studies, however, we propose to use probabilistic topic models to create "soft" clusters, further develop a full Bayesian model by integrating topic models with probabilistic MF.

Matrix Factorization with Topic Models. In the literature, researchers have made several attempts to combine topic models with MF, including CTM [Wang and Blei, 2011], HFT [McAuley and Leskovec, 2013], and ETF [Zhang *et al.*, 2014]. However, these methods mainly aim to incorporate ratings into topic models, and focus on combining the merits from both kinds of models. Typically, a single MF component is used, which is not suitable for local MF.

3 Bayesian Probabilistic Multi-Topic Matrix Factorization

In this section, we present our model BPMTMF for rating prediction. A glossary of notations used in the paper are listed in Table 1. In what follows, we denote matrices by bold capital letters. Superscripts, such as in $\mathbf{P}^{(k)}$, denote different topics' matrices for different superscripts; Subscripts on matrices denote the indices of data. For example, $\mathbf{R}_{um}$ denotes the entry in the u-th row and m-th column of the k-th topic matrix.

Table 1: Notations used in the paper.

Symbols	Descriptions
N, M	number of rows (users) and columns (items)
$\mathbf{R}$	data matrix ($\in \mathbb{R}^{N\times M}$) (with missing values)
K	the number ($\ll \min(N, M)$) of topics (or *topic number* for simplification)
D	the number ($\ll \min(N, M)$) of dimensions for latent vectors
$i = \langle u, m\rangle$	the i-th observation in $\mathbf{R}$
$\mathbf{P}_u^{(k)}$	the topic-specific latent vector ($\in \mathbb{R}^D$) for the u-th user *w.r.t.* the k-th topic
$\mathbf{Q}_m^{(k)}$	the topic-specific latent vector ($\in \mathbb{R}^D$) for the m-th item *w.r.t.* the k-th topic
z_i ($z_{u,m}$)	latent topic associated with observation $i = \langle u, m\rangle$
θ_u	topic distribution ($\in \mathbb{R}^K$) of the u-th user
ϕ_k	item distribution ($\in \mathbb{R}^M$) of the k-th topic
α	Dirichlet priors over topics for topic models
β	Dirichlet priors over items for topic models
Ψ_0	Gaussian-Wishart priors for probabilistic matrix factorization

3.1 The Proposed Model

Our main idea is to construct clusters over items using topic models, and predict the rating using the ensemble of multiple topic-specific probabilistic matrix factorizations. Hence, our

model consists of two parts: modeling the rated items and modeling the ratings.

Modeling the Rated Items. To model rated items, we adopt a similar approach following standard topic models (*e.g.,* Latent Dirichlet Allocation [Blei *et al.*, 2003]) by making the following analogy: an item is considered as a word token while the set of rated items by a user is considered as a document. In this way, a *topic* (*i.e.,* item topic) is defined as a multinomial distribution over the set of items. Let ϕ_k denote the k-th topic and $\phi_{k,m}$ denote the probability of the m-th item in the k-th topic. Given a set of K topics, the user preference is modeled as a multinomial distribution over them. Let θ_u denote the topical distribution of the u-th user and $\theta_{u,k}$ denote the probability of the k-th topic in the topical distribution of the u-th user. We employ symmetric Dirichlet priors $Dir(\alpha)$ and $Dir(\beta)$ (with the hyper-parameters of α and β) on θ and ϕ respectively. The topic modeling is used to generate the set of rated items for users as follows

$$P(\{\langle u,m\rangle\}) \propto \prod_{\langle u,m\rangle}\Big(\sum_k \theta_{u,k}\phi_{k,m}\Big), \tag{2}$$

where a pair $\langle u,m\rangle$ indicates that the u-th user has rated the m-th item. Eq. 2 enumerates all such pairs in the dataset.

Modeling the Ratings. To model ratings, topics are considered as contextual information, further a user or item is associated with a topic-specific latent vector for each topic. Let $\mathbf{P}_u^{(k)} \in \mathbb{R}^D$ (or $\mathbf{Q}_m^{(k)} \in \mathbb{R}^D$) denote the topic-specific latent vector for the u-th user (or the m-th item) *w.r.t.* the k-th topic. We assume that a user will reflect different rating behaviors in varying topical contexts, and an item will show different rated patterns by users in varying topical contexts. For example, a user is a fan of "Star War", who is likely to give high ratings to the movies of "Star Wars Episodes" but low ratings to other action movies. Note that simply incorporating a categorical bias like in [Mirbakhsh and Ling, 2013; Hu *et al.*, 2014] will not solve the issue in the above example, since the user give both high and low ratings in the same "Action" category. Our model tries to capture the *personal impact* of topical contexts on users' rating process. To draw topic-specific latent vectors, we employ Gaussian-Wishart priors on latent vectors $\mathbf{P}^{(k)}$ and $\mathbf{Q}^{(k)}$ with the parameters of $\Psi_{\mathbf{P}}^{(k)} = \{\mu_{\mathbf{P}}^{(k)}, \Lambda_{\mathbf{P}}^{(k)}\}$ and $\Psi_{\mathbf{Q}}^{(k)} = \{\mu_{\mathbf{Q}}^{(k)}, \Lambda_{\mathbf{Q}}^{(k)}\}$ as follows:

$$P(\Psi^{(k)}|\Psi_0^{(k)}) = \mathcal{N}(\mu^{(k)}|\mu_0^{(k)}, (\xi_0^{(k)}\Lambda^{(k)})^{-1})\mathcal{W}(\Lambda^{(k)}|\mathbf{W}_0^{(k)}, \nu_0^{(k)}) \tag{3}$$

where $\nu_0^{(k)}$ is the degrees of freedom of Wishart distribution $\mathcal{W}^{(k)}$, $\mathbf{W}_0^{(k)}$ is the scale matrix $\in \mathbb{R}^{N\times N}$ for user (or $\in \mathbb{R}^{M\times M}$ for item) and $\Psi_0^{(k)} = \{\mu_0^{(k)}, \nu_0^{(k)}, \mathbf{W}_0^{(k)}\}$ for the k-th topic. Given the k-th topic, a rating $\mathbf{R}_{um}$ is generated according to a Gaussian distribution

$$P(\mathbf{R}_{um}|\mathbf{P}_u^{(k)}, \mathbf{Q}_m^{(k)}, \sigma_k^2) = \mathcal{N}(\mathbf{R}_{um}|{\mathbf{P}_u^{(k)}}^\top \mathbf{Q}_m^{(k)}, \sigma_k^2), \tag{4}$$

1. For each topic $k = 1, ..., K$,
   (1) Draw a multinomial distribution $\phi_k \sim Dir(\beta)$
   (2) Draw the hyperparameters of the topic-specific user and item latent vectors $P(\Psi_{\mathbf{P}}^{(k)}|\Psi_0^{(k)})$ and $P(\Psi_{\mathbf{Q}}^{(k)}|\Psi_0^{(k)})$
2. For each item $m = 1, ..., M$,
   i. For each topic $k = 1,...,K$, draw the topic-specific item latent vector $\mathbf{Q}_m^{(k)} \sim P(\mathbf{Q}_m^{(k)}|\Psi_{\mathbf{Q}}^{(k)})$
3. For each user $u = 1, ..., N$,
   i. Draw $\theta_u \sim Dir(\alpha)$
   ii. For each topic $k = 1, ..., K$, draw the topic-specific user latent vector $\mathbf{P}_u^{(k)} \sim P(\mathbf{P}_u^{(k)}|\Psi_{\mathbf{P}}^{(k)})$
   iii. For each rated item m by u
      (1) Draw a topic $z \sim Disc(\theta_u)$
      (2) Draw $m \sim Disc(\phi_z)$
      (3) Draw the rating $\mathbf{R}_{um} \sim \mathcal{N}(\mathbf{R}_{um}|{\mathbf{P}_u^{(z)}}^\top \mathbf{Q}_m^{(z)}, \sigma_z^2)$

Figure 1: The generative process of the BPMTMF model.

where the mean and variance are set to ${\mathbf{P}_u^{(k)}}^\top \mathbf{Q}_m^{(k)}$ and σ_k^2 respectively.

The Final Model. Our proposed model, called BPMTMF, integrates the above two components, *i.e.,* the modeling of rated items (Eq. 2) and the modeling of ratings (Eq. 4), in a full Bayesian approach. We present the generative process of BPMTMF in Fig. 1. We use item topics to connect these two components, *i.e.,* using topic-specific latent vectors. The generative story can be described as follows. When the u-th user wants to rate the m-th item, she first draws a topic assignment of z according to her topical distribution θ_u. Then the m-th item is generated using ϕ_z. Finally the rating is generated based on the latent vectors $\mathbf{P}_u^{(k)}$ and $\mathbf{Q}_m^{(k)}$ corresponding to the k-th topic. Given the hyper-parameters, the likelihood over all ratings is as below

$$\begin{aligned} P(\mathbf{R}|\alpha,\beta,\Psi_0,\sigma) = & \\ \int \Big(\prod_u P(\theta_u|\alpha)\Big)\Big(\prod_k P(\phi_k|\beta)P(\Psi_{\mathbf{P}}^{(k)}|\Psi_0^{(k)})P(\Psi_{\mathbf{Q}}^{(k)}|\Psi_0^{(k)})\Big) & \\ \Big(\prod_k\prod_m P(\mathbf{Q}_m^{(k)}|\Psi_{\mathbf{Q}}^{(k)})\Big)\Big(\prod_k\prod_u P(\mathbf{P}_u^{(k)}|\Psi_{\mathbf{P}}^{(k)})\Big) & \\ \Big(\prod_{\langle u,m\rangle}\sum_k \theta_{u,k}\cdot\phi_{k,m}\cdot P(\mathbf{R}_{um}|\mathbf{P}_u^{(k)}, \mathbf{Q}_m^{(k)}, \sigma_k^2)\Big) & \\ \mathrm{d}\mathbf{P}_u^{(k)}\mathrm{d}\mathbf{Q}_m^{(k)}\mathrm{d}\Psi_{\mathbf{P}}^{(k)}\mathrm{d}\Psi_{\mathbf{Q}}^{(k)}\mathrm{d}\theta_u\mathrm{d}\phi_k. & \end{aligned} \tag{5}$$

Note that setting topic-specific latent vectors itself will increase the hazard of overfitting, while our Bayesian approach is effective to control the model complexity via using hyper-priors. Although we incorporate more hyper-parameters, as shown in BPMF [Salakhutdinov and Mnih, 2008] and our empirical results, the performance is relatively insensitive to the selection of hyper-parameters.

3.2 Model Learning with a Collapsed Gibbs Sampler

In our model, the parameters (or variables) to learn are listed as follows: users' topical distributions $\{\theta_u\}$, item topics $\{\phi_k\}$, users' latent vectors $\{\mathbf{P}_u^{(k)}\}$ and items' latent vectors $\{\mathbf{Q}_m^{(k)}\}$. Our task is to learn the parameters $\{\theta, \phi, \mathbf{P}, \mathbf{Q}\}$ to maximize the likelihood of observing rating matrix $\mathbf{R}$. It is difficult to directly optimize such an objective function due to the complex coupling of parameters and hidden variables. We adopt the commonly used Gibbs sampling algorithm [Andrieu *et al.*, 2003] for both inference and parameter learning. In each iteration, we alternatively infer topic assignments and update the topic-specific latent factors $\{\mathbf{P}_u^{(k)}\}$ and $\{\mathbf{Q}_m^{(k)}\}$. When the algorithm converges, we can estimate $\{\theta_u\}$ and $\{\phi_k\}$ using the topical assignments.

Inferring Topic Assignments. Fixing all topic-specific latent vectors and the hyper-parameters, we can derive the conditional distribution for a data entry $i = \langle u, m \rangle$ as below

$$\begin{aligned} &P(z_i = k | \mathbf{Z}_{\neg i}, \mathbf{R}, \mathbf{P}, \mathbf{Q}, \alpha, \beta, \sigma) \qquad (6) \\ \propto &\frac{n_u^k + \alpha - 1}{\sum_{j=1}^K (n_u^j + \alpha) - 1} \times \frac{n_m^k + \beta - 1}{\sum_{h=1}^M (n_h^k + \beta) - 1} \times \\ &\mathcal{N}(\mathbf{R}_{um} | {\mathbf{P}_u^{(k)}}^\top \mathbf{Q}_m^{(k)}, \sigma_k^2), \end{aligned}$$

where n_u^k denotes the number of items rated by the u-th user with the k-th topic, n_m^k denotes the number of users who rate the m-th item with the k-th topic, and the rating $\mathbf{R}_{um}$ is generated by the probability of $\mathcal{N}(\mathbf{R}_{um} | {\mathbf{P}_u^{(k)}}^\top \mathbf{Q}_m^{(k)}, \sigma_k^2)$ using the latent vectors corresponding to the k-th topic. Actually, the sampling equation is similar to the derivations for standard LDA models in [Heinrich, 2005] except that we incorporated the generation of ratings.

Updating Topic-specific Latent Vectors. The process of updating topic-specific latent factors is similar to the original Bayesian Probabilistic Matrix Factorization (BPMF) [Salakhutdinov and Mnih, 2008]. The difference lies in that we assume the topic assignment for each rated item is given, then the updating is performed on the corresponding topic-specific latent vectors. The conditional distribution over users' topic-specific latent vectors $\mathbf{P}_u^{(k)}$ is a Gaussian distribution:

$$\begin{aligned} &P(\mathbf{P}_u^{(k)} | \mathbf{R}, \mathbf{Q}^{(k)}, \Psi_{\mathbf{P}}^{(k)}, \sigma_k^2) \qquad (7) \\ &= \mathcal{N}(\mathbf{P}_u^{(k)} | {\mu_{\mathbf{P}}^{(k)}}^*, [{\Lambda_{\mathbf{P}}^{(k)}}^*]^{-1}) \\ &\propto P(\mathbf{P}_u^{(k)} | \mu_{\mathbf{P}}^{(k)}, \Lambda_{\mathbf{P}}^{(k)}) \prod_{m=1}^M \mathcal{N}(\mathbf{R}_{um} | {\mathbf{P}_u^{(k)}}^\top \mathbf{Q}_m^{(k)}, \sigma_k^2)^{\mathbf{I}_{um}^{(k)}}, \end{aligned}$$

in which we have

$${\Lambda_u^{(k)}}^* = \Lambda_{\mathbf{P}^{(k)}} + \frac{1}{\sigma_k^2} \sum_{m=1}^M (\mathbf{Q}_m^{(k)} {\mathbf{Q}_m^{(k)}}^\top)^{\mathbf{I}_{um}^{(k)}} \qquad (8)$$

$${\mu_u^{(k)}}^* = [{\Lambda_u^{(k)}}^*]^{-1} (\Lambda_{\mathbf{P}^{(k)}} \mu_{\mathbf{P}^{(k)}} + \frac{1}{\sigma_k^2} \sum_{m=1}^M (\mathbf{Q}_m^{(k)} \mathbf{R}_{um}))^{\mathbf{I}_{um}^{(k)}} \qquad (9)$$

where $\mathbf{I}_{um}^{(k)}$ is an indicator value which is equal to 1 only when the topic assignment $z_{u,m} = k$. We can learn items' latent vectors $\{\mathbf{Q}_m^{(k)}\}$ similarly, which are omitted here.

The Overall Learning Algorithm. In Alg. 1, we present the overall Gibbs sampling learning algorithm [Andrieu *et al.*, 2003] for the BPMTMF model. At the beginning, we use BPMF to initialize topic-specific latent vectors $\mathbf{P}$ and $\mathbf{Q}$. In each iteration, we first sample the topic assignments for all the observed data entries, then update $\mathbf{P}$ and $\mathbf{Q}$ with topic assignments fixed. After burn-in periods, we can estimate the parameters of $\{\theta_u\}$ and $\{\phi_k\}$ using a simple counting method for each iterative

$$\begin{aligned} \theta_{u,k} &= \frac{n_u^k + \alpha}{\sum_{j=1}^K (n_u^j + \alpha)}, \qquad (10) \\ \phi_{k,m} &= \frac{n_m^k + \beta}{\sum_{h=1}^M (n_h^k + \beta)}, \end{aligned}$$

where n_u^k, n_u^j, n_m^k and n_h^k are the counts defined in Eq. 6.

Computational Complexity Analysis. Let S be the total number of non-zero observations in $\mathbf{R}$, and $n_{u,\cdot}^k = \sum_m \mathbf{I}_{um}^{(k)}$ defined in Eq. 10. In an iteration, the running time for updating topic assignments is roughly $\mathcal{O}(KDS)$. For updating users' latent vectors, the cost mainly comes from the computation of $\sum_{m=1}^M (\mathbf{Q}_m^{(k)} {\mathbf{Q}_m^{(k)}}^\top)^{\mathbf{I}_{um}^{(k)}}$ (Eq. 8) and the matrix inverse for ${\Lambda_u^{(k)}}^*$ (Eq. 9), which take the costs of $\mathcal{O}(D^2 n_{u,\cdot}^{(k)})$ and $\mathcal{O}(D^3)$ respectively. Here, we assume that it requires $\mathcal{O}(D^3)$ time for the $\mathbb{R}^{D \times D}$ matrix inversion even though more efficient algorithms exist [Hu *et al.*, 2008]. Hence, updating ${\Lambda_u^{(k)}}^*$ and ${\mu_u^{(k)}}^*$ takes a cost of $\mathcal{O}(D^3 + D^2 n_{u,\cdot}^{(k)})$. Enumerating N users and K topics, it takes a cost of $\mathcal{O}(D^3 KN + D^2 S)$ with $S = \sum_{u,k} n_{u,\cdot}^k$. Similarly, updating items' latent vectors takes a cost of $\mathcal{O}(D^3 KM + D^2 S)$. To sum the above parts, the cost of an iteration is $\mathcal{O}(DKS + D^2 S + D^3 KN + D^3 KM)$.

3.3 Rating Predictions

When all the parameters of BPMTMF are learnt, we can use the following formula for rating prediction

$$\begin{aligned} &\hat{\mathbf{R}}_{um} \approx \qquad (11) \\ &\frac{1}{\sum_{k'=1}^K \theta_{u,k'} \cdot \phi_{m,k'}} \sum_{k=1}^K \left\{ (\theta_{u,k} \cdot \phi_{m,k}) ({\mathbf{P}_u^{(k)}}^\top \mathbf{Q}_m^{(k)}) \right\}, \end{aligned}$$

where $\hat{\mathbf{R}}_{um}$ is the predicted value for $\mathbf{R}_{um}$. The above prediction uses only a single Gibbs sample, which can be simply extended to average the results from multiple samples.

Connections with Previous Studies. Eq. 1 presents a general formulation for previous studies on non-probabilistic local matrix factorization [Mackey *et al.*, 2011; Lee *et al.*, 2013; Chen *et al.*, 2015]. Interestingly, our prediction formula

Algorithm 1: The learning algorithm for BPMTMF.

Use BPMF to initialize $\mathbf{P}^{(k)}$ and $\mathbf{Q}^{(k)}$;
repeat
  for *each entry* $i = \langle u, m \rangle$ **do**
    Sample a topic assignment z_i using Eq. 6:

$$z_i = P(z_i = k | \mathbf{Z}_{\neg i}, \mathbf{R}, \mathbf{P}, \mathbf{Q}, \alpha, \beta, \sigma)$$

  end
  for *each topic* $k = 1, 2, ..., K$ **do**
    Sample Gaussian-Wishart priors using Eq. 3:

$$\Psi_{\mathbf{P}}^{(k)} = P(\Psi_{\mathbf{P}}^{(k)} | \Psi_0^{(k)}),\ \Psi_{\mathbf{Q}}^{(k)} = P(\Psi_{\mathbf{Q}}^{(k)} | \Psi_0^{(k)})$$

    for *each user* $u = 1, 2, ..., N$ **do**
      Sample users' topic-specific latent vectors using Eq. 7:

$$\mathbf{P}_u^{(k)} = P(\mathbf{P}_u^{(k)} | \mathbf{R}, \mathbf{Q}^{(k)}, \Psi_{\mathbf{P}}^{(k)}, \sigma_k)$$

    end
    for *each item* $m = 1, 2, ..., M$ **do**
      Sample items' topic-specific latent vectors:

$$\mathbf{Q}_m^{(k)} = P(\mathbf{Q}_u^{(k)} | \mathbf{R}, \mathbf{P}^{(k)}, \Psi_{\mathbf{Q}}^{(k)}, \sigma_k)$$

    end
  end
until *Convergence*;

(Eq. 11) has a close connection with Eq. 1. Given a pair of $\langle u, m \rangle$, we can have the corresponding mappings: ${\mathbf{P}_u^{(k)}}^\top \mathbf{Q}_m^{(k)} \rightarrow \hat{\mathbf{R}}_{um}^{(k)}$ indicating the predicted value for $\mathbf{R}_{um}$ in the k-th "cluster", $\theta_{u,k} \cdot \phi_{m,k} \rightarrow \mathbf{L}_{um}^{(k)}$ indicating the weight for $\hat{\mathbf{R}}_{um}^{(k)}$, and $\sum_{k'=1}^{K} \theta_{u,k'} \cdot \phi_{m,k'} \rightarrow \mathbf{Z}_{um}$ playing the role of the normalizer. With such an analogy, our formula can be considered as a probabilistic realization of previous non-probabilistic methods: a cluster in Eq. 1 is essentially a topic in our approach. Such a connection sheds lights on that previous studies can be explained in a probabilistic way with more deep theoretical analysis and extensions.

4 Experiments and Analysis

4.1 Experimental Setup

Datasets. We evaluate the models on the two publicly available movie datasets Movielens[1] and Netflix[2] described in Table 2. We randomly split the data into training set and testing set with the ratio $9:1$. The final results are reported by the average of five such runs.

Table 2: Statistics of our datasets.

Data Set	#Users	#Items	#Ratings	Density
Movielens	69,878	10,677	10,000,054	1.31%
Netflix	480,189	17,770	100,000,000	1.17%

[1]http://www.grouplens.org/
[2]http://www.netflixprize.com/

Evaluation Metrics. We adopt the commonly used Root Mean Square Error (RMSE) to evaluate the predictive accuracy, defined by:

$$RMSE = \sqrt{\frac{\sum_{\langle u,m \rangle} (\mathbf{R}_{um} - \hat{\mathbf{R}}_{um})^2}{N_{test}}}$$

where N_{test} is the number of ratings in test set. A smaller value of RMSE indicates a better performance.

Comparison Methods. We compare the proposed BPMTMF with following baselines.

- DFC [Mackey *et al.*, 2011]: divides a large-scale matrix factorization task into smaller subproblems, and uses the techniques from randomized matrix approximation to combine the subproblem solutions.
- LLORMA [Lee *et al.*, 2013]: uses non-parametric kernel smoothing to develop local low-rank approximation and aggregate several submatices into unified matrix approximation.
- WEMAREC [Chen *et al.*, 2015]: constructs submatrices via partitional co-clustering and proposes a submatrix-based weighting strategy to predict the final ratings.
- PMTMF: As a direct comparison of our Bayesian approach, we also implement a non-Bayesian Probabilistic Multi-Topic Matrix Factorization, which does not have any prior parameters. Expectation-Maximization algorithm is employed for optimization.

We do not use traditional MF methods [Mnih and Salakhutdinov, 2007; Koren *et al.*, 2009] as baselines here, since previous studies [Mackey *et al.*, 2011; Lee *et al.*, 2013; Chen *et al.*, 2015] have shown that the above local MF methods have outperformed the traditional MF methods. We implement our methods PMTMF and BPMTMF based on librec toolkit [Guo *et al.*, 2015].

Parameter Setting. Following [Chen *et al.*, 2015; Lee *et al.*, 2013], the number of dimensions for latent vectors, *i.e.*, D, is set to 20 for all the methods. For BPMTMF and PMTMF, we empirically set the number of topics K to 20; Following [Griffiths and Steyvers, 2004], α and β are set to $\frac{50}{K}$ and 0.01 respectively. Following [Salakhutdinov and Mnih, 2008], we initialize $\mu_0^{(k)} = 0$, $\nu_0^{(k)} = D$, $\mathbf{W}_0^{(k)}$ to the identity matrix and variance $\sigma_k^2 = 2$. As indicated in [Salakhutdinov and Mnih, 2008], the predictive accuracy becomes relatively stable when the iteration number is larger than 150 for BPMF. Hence, for BPMTMF, we discard 200 iterations for burn-in, and run another 150 iterations for sampling. The other parameters for baselines are set to the reported optimal values in the original papers since we have similar experimental setup.

4.2 Results and Analysis

Performance Comparison for Rating Prediction. We present the performance of the comparison methods for rating prediction in Table 4. Among all the baseline methods, the

Table 3: Top ten items in a cluster or a topic from two sample movie genres on MovieLens dataset. A "√" indicates a correct categorization of an item in the corresponding category.

Action		**Drama**	
LLORMA	BPMTMF	LLORMA	BPMTMF
√ Star Wars: Episode IV	√ Indiana Jones and the Last Crusade	√ American Beauty	√ Silence of the Lambs
√ Star Wars: Episode V	√ Raiders of the Lost Ark	√ Braveheart	√ Saving Private Ryan
× American Beauty	√ Die Hard	√ Saving Private Ryan	√ Shawshank Redemption
× Shakespeare in Love	√ Star Wars: Episode IV	× L.A. Confidential	√ American Beauty
√ Saving Private Ryan	√ Terminator	× Star Wars: Episode IV	√ Pulp Fiction
× E.T. the Extra-Terrestrial	√ Star Wars: Episode V	√ Silence of the Lambs	√ Good Will Hunting
× Being John Malkovich	√ Batman	× Fugitive	√ Fargo
× Sixth Sense	√ Star Wars: Episode VI	× Star Wars: Episode VI	× Sixth Sense
√ Men in Black	√ Indiana Jones and the Temple of Doom	√ Schindler's List	× Forrest Gump
√ Star Wars: Episode VI	√ Hunt for Red October	× Toy Story	√ Braveheart

recently proposed WEMAREC performs best. WEMAREC adopts a partitional co-clustering method for submatrix generation, representing the state-of-art for local MF. Then we examine the performance of our proposed methods, namely PMTMF and BPMTMF. We can observe that PMTMF is only slightly worse than the best baseline WEMAREC. It indicates that our probabilistic realization of local MF (*i.e.,* PMTMF) can achieve very good performance compared with previous local MF methods. Further, the Bayesian model BPMTMF is substantially better than PMTMF and WEMAREC, which shows the effectiveness of the Bayesian approach. Note that the difference between PMTMF and BPMTMF is that BPMTMF implements PMTMF in a full Bayesian approach. Hence, to explain why BPMTMF improves over PMTMF, an important reason is that a Bayesian model is more effective to control the model complexity. Compared to the baseline methods, BPMTMF provides a more principled solution to both submatrix generation and weight setting in a joint model. It benefits from the Bayesian approach that fewer efforts are required in parameter selection due to the incorporation of hyper-priors. As a comparison, WEMAREC needs to set the co-clustering parameter and combine weights with more care.

Table 4: Comparison of RMSE results for different methods.

Methods	MovieLens	Netflix
DFC	0.8064	0.8451
LLORMA	0.7834	0.8243
WEMAREC	0.7769	0.8142
PMTMF	0.7792	0.8198
BPMTMF	**0.7679**	**0.8081**

Cluster Analysis. Besides performance improvement, another merit of our proposed model is that we characterize clusters as topics, which is likely to have more coherent semantics. To see this, we construct a qualitative analysis for the clustering results. Among the baselines, DFC adopts a randomized method for submatrix generation, and WEMAREC tends to produce very small submatrices, hence both are not suitable for clustering analysis. We select LLORMA as a comparison for BPMTMF. In MoiveLens dataset, a movie is attached with a set of movie genre labels. We first run both LLORMA and BPMTMF to produce a set of clusters (or topics). Given a learnt cluster (or topic), we assign a movie genre label with the most number of occurrences in the labels of its ten top representative items. Table 3 presents the two sample clusters generated by LLORMA and BPMTMF. We can observe that BPMTMF is able to generate more coherent clusters, *i.e.,* topics. With coherent topical semantics, it provides a more interpretable latent representation for users and items.

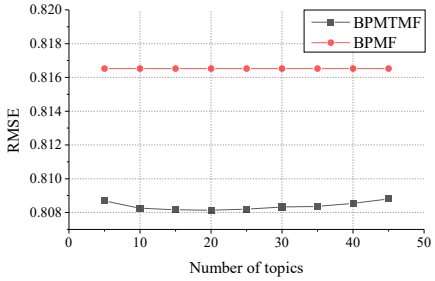

Figure 2: Performance with different numbers of topics on Netflix dataset.

Impact of the Topic Number. An important parameter in BPMTMF is the number of topics (*i.e.,* K), since we set topic-specific latent representations corresponding to each topic. We vary K in the interval $[5, 45]$ with a gap of 5, and report the RMSE performance of BPMTMF in Fig. 2. We can observe that the performance achieves the best when $15 \leq K \leq 25$. We also incorporate the performance of BPMF as a comparison, which is a special case of our model BPMTMF when $K = 1$. The performance when $K \geq 5$ is substantially better than that of $K = 1$. By combining the analysis for Table 3, we can see using multi-topic MF is effective to capture the complex characteristics of users and items, further improve the performance of rating prediction.

5 Conclusion

In this paper, we present a Bayesian formulation of local matrix factorization. Our approach elegantly integrates probabilistic matrix factorization with topic models in a joint model. We have also shown that there is a close connection between our model and previous non-probabilistic methods. Our model is more powerful to reflect the complex characteristics for users and items in rating prediction, and further

enhances the model interpretability. Extensive experiments on large real-world datasets demonstrate the effectiveness of the proposed model.

Acknowledgments This work was partially supported by NSFC grants (No. 61321064, 61472141, 61502502), Shanghai Knowledge Service Platform Project(No. ZF1213) and Beijing Natural Science Foundation (No. 4162032).

References

[Andrieu *et al.*, 2003] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1-2):5–43, 2003.

[Beutel *et al.*, 2015] Alex Beutel, Amr Ahmed, and Alexander J Smola. Accams: Additive co-clustering to approximate matrices succinctly. In *Proceedings of the 24th International Conference on World Wide Web*, pages 119–129. International World Wide Web Conferences Steering Committee, 2015.

[Blei *et al.*, 2003] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[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 *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 303–312. ACM, 2015.

[Dhillon *et al.*, 2003] Inderjit S Dhillon, Subramanyam Mallela, and Dharmendra S Modha. Information-theoretic coclustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 89–98. ACM, 2003.

[Griffiths and Steyvers, 2004] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[Guo *et al.*, 2015] Guibing Guo, Jie Zhang, Zhu Sun, and Neil Yorke-Smith. Librec: A java library for recommender systems. In *Posters, Demos, Late-breaking Results and Workshop Proceedings of the 23rd International Conference on User Modeling, Adaptation and Personalization*, 2015.

[Heinrich, 2005] Gregor Heinrich. Parameter estimation for text analysis. Technical report, Technical report, 2005.

[Hu *et al.*, 2008] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. IEEE, 2008.

[Hu *et al.*, 2014] Longke Hu, Aixin Sun, and Yong Liu. Your neighbors affect your ratings: on geographical neighborhood influence to rating prediction. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 345–354. ACM, 2014.

[Koren *et al.*, 2009] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[Lee *et al.*, 2013] Joonseok Lee, Seungyeon Kim, Guy Lebanon, and Yoram Singer. Local low-rank matrix approximation. In *Proceedings of The 30th International Conference on Machine Learning*, pages 82–90, 2013.

[Lee *et al.*, 2014] Joonseok Lee, Samy Bengio, Seungyeon Kim, Guy Lebanon, and Yoram Singer. Local collaborative ranking. In *Proceedings of the 23rd international conference on World wide web*, pages 85–96. ACM, 2014.

[Mackey *et al.*, 2011] Lester W Mackey, Michael I Jordan, and Ameet Talwalkar. Divide-and-conquer matrix factorization. In *Advances in Neural Information Processing Systems*, pages 1134–1142, 2011.

[McAuley and Leskovec, 2013] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM, 2013.

[Mirbakhsh and Ling, 2013] Nima Mirbakhsh and Charles X Ling. Clustering-based factorized collaborative filtering. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 315–318. ACM, 2013.

[Mnih and Salakhutdinov, 2007] Andriy Mnih and Ruslan Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2007.

[Paterek, 2007] Arkadiusz Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, volume 2007, pages 5–8, 2007.

[Salakhutdinov and Mnih, 2008] Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880–887. ACM, 2008.

[Shan and Banerjee, 2008] Hanhuai Shan and Arindam Banerjee. Bayesian co-clustering. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 530–539. IEEE, 2008.

[Wang and Blei, 2011] Chong Wang and David M Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456. ACM, 2011.

[Zhang *et al.*, 2014] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 83–92. ACM, 2014.