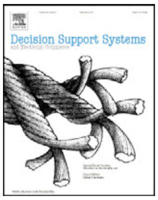
Decision Support Systems xxx (2013) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss



Bayesian Probabilistic Matrix Factorization with Social Relations and Item Contents for recommendation

Juntao Liu [a,b], Caihua Wu [c], Wenyu Liu [a,*]

[a] Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
[b] Department of Computer Engineering, Mechanical Engineering Institute, Shijiazhuang 050003, China
[c] Information Combat Commanding Teaching and Research Section, Information Countermeasure Department, Air Force Radar Academy, Wuhan 430010, China

ARTICLE INFO

Article history:
Received 3 September 2012
Received in revised form 27 March 2013
Accepted 4 April 2013
Available online xxxx

Keywords:
Recommendation system
Collaborative filtering
Social network
Item contents
Matrix factorization
Tags

ABSTRACT

Recommendation systems have received great attention for their commercial value in today's online business world. However, most recommendation systems encounter the data sparsity problem and the cold-start problem. To improve recommendation accuracy in this circumstance, additional sources of information about the users and items should be incorporated in recommendation systems. In this paper, we modify the model in Bayesian Probabilistic Matrix Factorization, and propose two recommendation approaches fusing social relations and item contents with user ratings in a novel way. The proposed approach is computationally efficient and can be applied to trust-aware or content-aware recommendation systems with very large dataset. Experimental results on three real world datasets show that our method gets more accurate recommendation results with faster converging speed than other matrix factorization based methods. We also verify our method in cold-start settings, and our method gets more accurate recommendation results than the compared approaches.

1. Introduction

Recommendation systems have become an important research area in the past decade. Recommendation systems typically try to predict the interests of a user by collecting rating information of other users or items. Recommendation methods are generally divided into collaborative filtering (CF) methods and content-based (CB) methods [19]. In content-based recommendation methods, the rating of an item for a user is estimated based on the ratings of similar items for this user. Collaborative filtering methods try to predict the rating of an item for a particular user based on the previous ratings of this item rated by other similar users. The underlying assumption of collaborative filtering is that similar users have similar tastes. Collaborative filtering methods are widely used in large commercial systems, such as Amazon and Netflix.

Matrix factorization is one of the most popular collaborative filtering methods in recent years. It is assumed that the preference of a user can be represented by a small number of unobserved features. Formally, supposing that there are M users and N items, $M \times N$ matrix R is the observed rating matrix. Matrix factorization based methods find $M \times D$ user latent feature matrix U and $N \times D$ item latent feature matrix V to minimize the loss function $f_{loss}\left(R, \hat{R}\right)$, which measures the difference between the observed rating matrix R and the predicted rating matrix $\hat{R} = UV^T$. Here D is the dimension of user feature vector and item feature vector. D is much less than M and N.

A number of algorithms have been proposed to solve matrix factorization for recommendation, such as Variational Bayesian Matrix Factorization (VBMF) [10], Probabilistic Matrix Factorization (PMF) [22], Bayesian Probabilistic Matrix Factorization (BPMF) [23], General Probabilistic Matrix Factorization (GPMF) [25] and so on. Same as other collaborative filtering methods, these methods encounter the data sparsity problem [2]. For a particular recommendation system, the density of the observed rating matrix is usually less than 1% [24]. In this case, it is difficult to find similar users or similar items. Another well-known problem in recommendation system is the cold-start problem, that is, how to provide recommendation to new users who have expressed very few ratings. It is believed that social relations among users can alleviate these problems. For example, in our daily life, we often turn to our friends for recommendations. We share interests with close friends. Our tastes are also often affected by our friends. Cooperating with social relations in recommendation systems can improve recommendation accuracy [15]. In recent years, some recommendation methods fusing social relations by regularization [6,9,16,28] or factorization [13,15] were proposed. On the other hand, item contents, such as tags, categories and item profile, also provide huge opportunity to improve the accuracy of recommendation. For example, we may like movies performed by a same actor.

In this paper, to alleviate the data sparsity problem and the cold-start problem and to improve recommendation accuracy further, we integrate social relations and item contents into the framework

[*] Corresponding author. Tel.: +86 27 87543236.
E-mail addresses: prolay@163.com (J. Liu), wucaihua_1999@yahoo.com.cn (C. Wu), liuwy@hust.edu.cn (W. Liu).

0167-9236/$ – see front matter. Crown Copyright © 2013 Published by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.dss.2013.04.002

of Bayesian Probabilistic Matrix Factorization (BPMF) [23] in a novel way which is different from regularization-based methods and factorization-based methods, and propose two novel recommendation methods cooperating with social relations and item contents. In BPMF, the hyperparameters of user feature vectors are the same for all users. But in practice, users' preferences vary greatly, so the hyperparameters of different users should be different. Generating user feature vectors by uniform user hyperparameters in BPMF may lead to some recommendation errors. In this paper we modify the model in BPMF, and suggest that the hyperparameters of different user vectors are different. We then propose the Bayesian Probabilistic Matrix Factorization with Social Relations (BPMFSR) recommendation method. To alleviate data sparsity problem and cold-start problem we fuse social relations in this method. We argue that the posterior distribution of user hyperparameters should be conditioned on the feature vectors of trusted users. The underlying assumption is that the users' preference may be influenced by their friends. On the other hand, similar to the uniform user hyperparameters, uniform item hyperparameters in BPMF also lead to some recommendation errors. To address this problem, we extend our BPMFSR method by fusing item contents information and propose an improved algorithm, Bayesian Probabilistic Matrix Factorization with Social Relations and Item Contents (BPMFSRIC). In BPMFSRIC, we assume that the hyperparameters of different item vectors are different, and the posterior distribution of item feature vector parameters is conditioned on the feature vectors of linked items. The links among items can be extracted by item contents information, such as tags, categories and properties. The BPMFSR can be applied to trust-aware recommender systems. If item contents are known additionally, BPMFSRIC can improve recommendation accuracy further. The proposed method is computationally efficient, and can be applied to large-scale real life datasets. Experimental results on Douban dataset [16], Epinions dataset [17] and Last.fm dataset [4] show that the accuracy of our method outperforms other methods based on matrix factorization. We also verify our method in cold-start settings, and our method gets better results than other methods.

The rest of this paper is organized as follows. In Section 2, a survey of major recommendation methods based on matrix factorization is provided. Section 3 introduces PMF and BPMF briefly. The proposed method is described in Section 4. The experimental results are presented and analyzed in Section 5 followed by the conclusions and further work in Section 6.

2. Related work

Rating-based recommendation methods are generally divided into collaborative filtering (CF) methods and content-based (CB) methods [19]. Matrix factorization based methods are one kind of collaborative filtering methods. In this section we review several recommendation methods based on matrix factorization.

Lim and Teh [10] proposed Variational Bayesian based Matrix Factorization (VBMF) for movie recommendation. Nakajima and Sugiyama [18] analyzed VBMF theoretically. Probabilistic Matrix Factorization (PMF) proposed by Salakhutdinov and Mnih [22] models the predictive error of matrix factorization as Gaussian distribution. Gradient descent algorithm is used to find the local maximal of the posterior probability over user and item latent matrices with parameters. PMF gets accurate results on Netflix dataset. The main shortcoming of PMF is that careful parameter tuning is needed to avoid over fitting. This leads to high computational complexity on large datasets. Bayesian Probabilistic Matrix Factorization (BPMF) [23] overcomes this drawback by using Markov Chain Monte Carlo (MCMC) method that gets more accurate predictive results. As far as we know, BPMF method outperforms most of the recommendation methods based on matrix factorization. Shan and Banerjee [5] extended PMF and BPMF and proposed a series of general PMF (GPMF) methods. Porteous et al. [20] fused side information into BPMF model. In their model, the ratings are estimated by the product of the user latent matrix and the item matrix and the regression of user and item side information. Adams et al. [1] modified the BPMF model. The observed rating matrix, the user latent matrix and the item latent matrix are represented by time-varying functions. The variation processes of user latent matrix and item matrix are represented by Gaussian processes. Lu et al. [11] proposed a matrix factorization based recommendation method to predict the variation of ratings with time. Two regularization terms, spatial term and temporal term, are added into the objective function. Gemulla et al. [5] proposed a stratified stochastic gradient descent (SSGD) algorithm to solve the general matrix factorization problem, and gave sufficient conditions for convergence. Luo et al. [12] proposed an incremental collaborative filtering recommendation method based on the regularized matrix factorization.

Generally, the main challenges for recommendation systems are the data sparsity problem and the cold-start problem [2]. To address these problems, in recent years, researchers proposed some matrix factorization based recommendation methods fusing social relations among users with rating data, which can help to improve the performance of recommender systems. These methods can be divided into two types: regularization-based methods and factorization-based methods.

Regularization-based methods typically add regularization term to the loss function and minimize it. For examples, recommendation method proposed by Hao Ma et al. [16] adds social regularization term to the loss function, which measures the difference between the latent feature vector of a user and those of his (or her) friends. Local minimum of the loss function is found by gradient-based method. Jamali and Ester [6] proposed a probability model similar to the model 1 in [16]. Relation regularized matrix factorization (RRMF) [9] method proposed by Li and Yeung adds the graph Laplacian regularization term of social relations into the loss function and minimizes the loss function by alternative projection algorithm. Zhu et al. [28] used the same model in [9] and built graph Laplacian of social relations using three kinds of kernel functions. The minimization problem is formulated as low-rank semidefinite program (LRSDP) and is solved by the method proposed in [3]. Regularization-based methods always minimize the difference between the latent feature vector of a user and those of his (or her) friends and give weights to the regularization terms to tradeoff between factorization error and regularization terms. The weights should be tuned manually to avoid over fitting. So the drawback of this kind of methods is the same as that of PMF.

In factorization-based methods, social relations are represented as social relation matrix, which is factored as well as the rating matrix. The loss function is the weighted sum of the social relation matrix factorization error and the rating matrix factorization error. For example, SoRec [13] factorizes social relation matrix and rating matrix simultaneously. Social relation matrix is approximated as the product of the user latent feature matrix and the factor feature matrix. SoRec can also be extended to fuse social tags and item tags with rating information [15]. Yuan et al. [27] argued that factorization-based methods outperform regularization-based methods for fusing membership information, and proposed a method fusing membership and friendship by factorization and regularization, respectively. Factorization-based methods encounter the same problem as regularization-based methods: the weight of the social relation matrix factorization error and the weight of the rating matrix factorization error should be tuned to avoid over fitting, which is computationally expensive especially on large-scale datasets. To avoid parameter tuning, Singh and Gordon proposed Hierarchical Bayesian Collective Matrix Factorization (HBCMF) [26], in which two relation matrices are factored. The model of HBCMF is very similar to that of BPMF, and a block Metropolis–Hastings algorithm is used to sample from this model.

In this paper, we propose two novel recommendation methods cooperating with social relations and item contents. Our methods are different from previous methods because the way we use social relations and item contents is not factorization-based and regularization-based. To fuse social relations and item contents, we modify the model in BPMF. The differences between our methods and PMF, and BPMF are shown in Fig. 1. Our methods not only avoid parameter tuning just as BPMF, but also improve recommendation accuracy and converge speed.

3. Preliminaries

In this section, we first introduce preliminaries for matrix factorization based recommendation method. And then, we introduce the frameworks of PMF and BPMF briefly.

3.1. Matrix factorization for recommendation

Suppose there are M users and N items. Let matrix R denote the rating matrix, where R_{ij} represents the rating of user i for item j. In most online systems, R_{ij} is the K-point integer. For example, in Douban website (http://www.douban.com), rating values are 5-point integers, 1-point means 'very bad' and 5-point means 'excellent'. Let $U \in \mathbb{R}^{M \times D}$ and $V \in \mathbb{R}^{N \times D}$ be the user and item latent feature matrices, where row vectors U_i and V_j represent user-specific and item-specific latent feature vectors. D is the dimension of user feature vector and item feature vector, which is much less than M and N. In Probabilistic Matrix Factorization (PMF) [22] method, the conditional probability of observed ratings matrix R is modeled as:

$$p\left(R|U,V,\sigma^2\right) = \prod_{i=1}^{M}\prod_{j=1}^{N}\left[N\left(R_{ij}|U_iV_j^T,\sigma^2\right)\right]^{I_{ij}} \tag{1}$$

where $N(x|\mu, \sigma^2)$ is the probability density function of the Gaussian distribution with mean μ and variance σ^2. I is the indicator matrix. I_{ij} is equal to 1 if user i rated item j and 0 otherwise. The prior distributions of latent matrices U and V are modeled as:

$$p\left(U|\sigma_U^2\right) = \prod_{i=1}^{M} N\left(U_i|0,\sigma_U^2\right) \tag{2}$$

$$p\left(V|\sigma_V^2\right) = \prod_{i=1}^{N} N\left(V_j|0,\sigma_V^2\right). \tag{3}$$

The graphical model for PMF is shown in Fig. 1(a). This model is learned by maximizing posterior probability of latent matrices U and

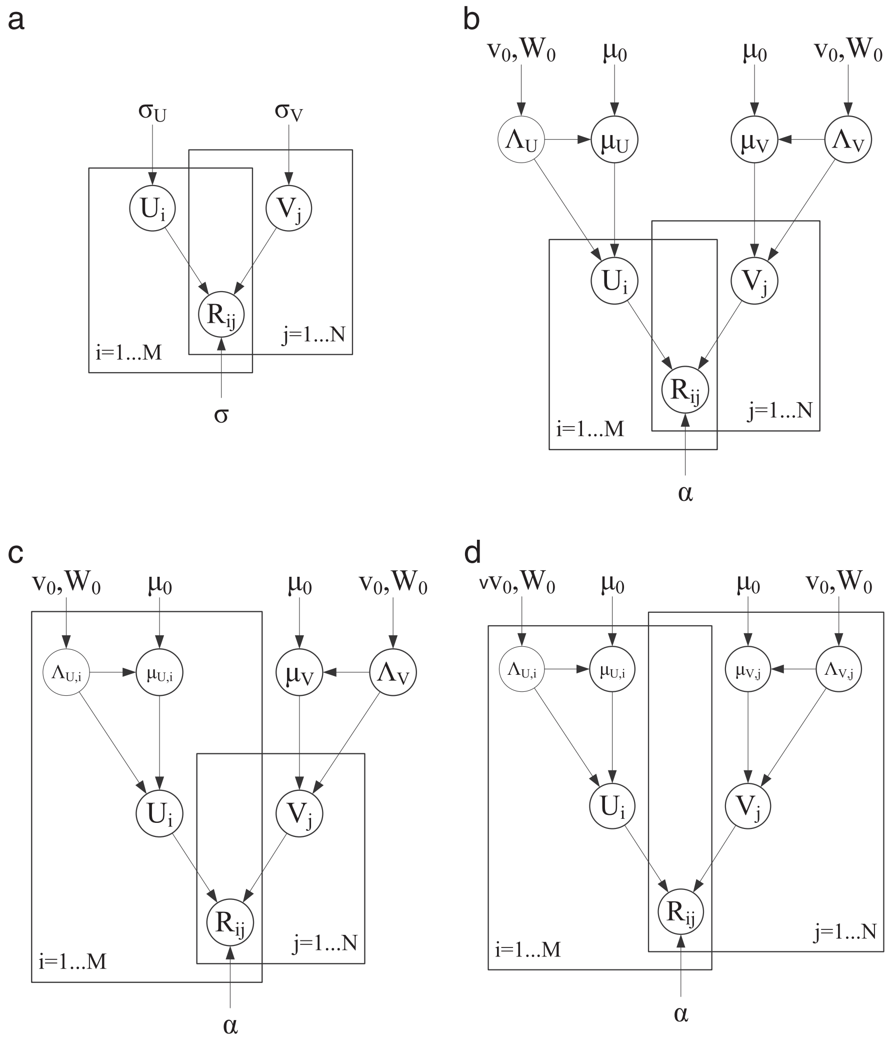


Fig. 1. Graphical models for PMF (a), BPMF (b), BPMFSR (c) and BPMFSRIC (d). The main difference between BPMF and BPMFSR is that BPMFSR generates user hyperparameters separately for every user vectors, while BPMF uses uniform user hyperparameters. BPMFSRIC not only generates user hyperparameters separately but also generate item hyperparameters separately.

V, which is equivalent to minimizing sum-of-squares of factorization error with quadratic regularization terms [22]:

$$E = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N} I_{ij}\left(R_{ij} - U_i V_j^T\right)^2 + \frac{\lambda_U}{2}\|U\|_{Fro}^2 + \frac{\lambda_V}{2}\|V\|_{Fro}^2 \tag{4}$$

where $\lambda_U = \sigma^2/\sigma_U^2$, $\lambda_V = \sigma^2/\sigma_V^2$ and $\|\cdot\|_{Fro}^2$ denote Frobenius norm. Local minimum of Eq. (4) is found through gradient descent method in PMF [22].

Although PMF is maybe the most popular method for collaborative filtering and it is very successful in the Netflix Prize contest, the drawbacks of this method are two-fold. Firstly, it requires careful tuning of parameters to avoid over fitting. This process is computationally expensive on large datasets. Secondly, PMF assumes that user vectors and item vectors are independent and identically distributed and ignores the social relations among users. It is believed that social relations can alleviate the data sparsity problem, and improve recommendation accuracy. For this reason, Hao Ma et al. [16] add social regularization term to the loss function in Eq. (4). The social regularization term measures the difference between the feature vector of a user and those of his (or her) friends. Hao Ma's method gets more accurate recommendation results than NMF [8], PMF [22], RSTE [14] and other state-of-art on large real life datasets. In this paper, we compare our methods with Hao Ma's method.

3.2. Bayesian Probabilistic Matrix Factorization

To avoid parameter tuning, Salakhutdinov and Mnih [23] proposed a Bayesian model of Probabilistic Matrix Factorization (BPMF). The prior distributions of latent matrices U and V are given by:

$$p\left(U|\mu_U, \Lambda_U\right) = \prod_{i=1}^{M} N\left(U_i|\mu_U, \Lambda_U^{-1}\right) \tag{5}$$

$$p\left(V|\mu_V, \Lambda_V\right) = \prod_{i=1}^{N} N\left(V_j|\mu_V, \Lambda_V^{-1}\right). \tag{6}$$

BPMF assumes that user hyperparameters $\Theta_U = \{\mu_U, \Lambda_U\}$ and item hyperparameters $\Theta_V = \{\mu_V, \Lambda_V\}$ follow Gaussian-Wishart distribution. The prior distributions of user hyperparameters and item hyperparameters are given by:

$$p(\Theta_U|\Theta_0) = p(\mu_U|\Lambda_U)\, p\,(\Lambda_U) = N(\mu_U|\mu_0, \left(\beta_0\Lambda_U\right)^{-1})W(\Lambda_U|W_0, v_0) \tag{7}$$

$$p(\Theta_V|\Theta_0) = p(\mu_V|\Lambda_V)\, p\,(\Lambda_V) = N\left(\mu_V|\mu_0, (\beta_0\Lambda_V)^{-1}\right)W(\Lambda_V|W_0, v_0) \tag{8}$$

where $W(w|W_0, v_0)$ is the Wishart distribution with v_0 degrees of freedom and a $D \times D$ scale matrix W_0. $\Theta_0 = \{\mu_0, v_0, W_0\}$. The graphical model for BPMF is shown in Fig. 1(b). BPMF uses Gibbs algorithm to sample user and item feature vectors. It is assumed that the posterior distribution over user feature vector U_i, which is conditioned on item feature matrix V, observed rating matrix R and hyperparameters, is Gaussian:

$$\begin{aligned} &p(U_i|R, V, \Theta_U, \alpha) = N\left(U_i|\mu_i^*, (\Lambda_i^*)^{-1}\right) \\ &\sim\prod_{j=1}^{N}\left[N\left(R_{ij}|U_i^T V_j, \alpha^{-1}\right)\right]^{I_{ij}} p(U_i|\mu_U, \Lambda_U) \end{aligned} \tag{9}$$

where $\Lambda_i^* = \Lambda_U + \alpha\sum{}_{j=1}^{N} V_j^T V_j)$ and $\mu_i^* = (\Lambda_i^*)^{-1}\left(\alpha\sum_{j=1}^{N}(V_j R_{ij})^{I_{ij}} + \mu_U\Lambda_U\right)$.

The conditional distribution over user hyperparameters conditioned on the user feature matrix U is given as:

$$p(\mu_U, \Lambda_U|U, \Theta_0) = N\left(\mu_U|\mu_0^*, (\beta_0^*\Lambda_U)^{-1}\right)W(\Lambda_U|W_0^*, v_0^*) \tag{10}$$

where

$$\mu_0^* = \frac{\beta_0\mu_0 + M\overline{U}}{\beta_0 + M}, \beta_0^* = \beta_0 + M, v_0^* = v_0 + M, \overline{S} = \tfrac{1}{M}\sum_{i=1}^{M}(U_i - \overline{U})^T(U_i - \overline{U})$$

$$\overline{U} = \tfrac{1}{M}\sum_{i=1}^{M} U_i \text{ and } W_0^* = \left({W_0}^{-1} + M\overline{S} + \frac{\beta_0 M}{\beta_0 + M}(\mu_0 - \overline{U})^T(\mu_0 - \overline{U})\right)^{-1}.$$

BPMF gets more accurate recommendation results than PMF, and avoids parameter tuning. But BPMF also ignores social relations among users. The distribution parameters of user feature vectors are estimated by all of the user feature vectors. It is observed that people with social relations are more likely to share same preferences. If we estimate distribution parameters of a particular user feature vector by the vectors of his (or her) friends, the recommendation accuracy will be improved further. By this idea, we propose our recommendation methods.

4. Proposed method

In BPMF framework, hyperparameters for all users are the same (see Eq. (5)). This is unreasonable because users' preferences are different, and the hyperparameters should be different too. We assume that user hyperparameters are different for different users, and propose Bayesian Probabilistic Matrix Factorization with Social Relations (BPMFSR), in which user hyperparameters are sampled according to the social relations. Our method uses the social relations in a novel way, which is not regularization-based and factorization-based, and can improve recommendation accuracy.

In BPMF, uniform item hyperparameters encounter the same problem. To improve the performance further, we fuse item contents as well as social relations, and propose Bayesian Probabilistic Matrix Factorization with Social Relations and Item Contents (BPMFSRIC), in which item hyperparameters are sampled according to the item contents.

In this section, we introduce BPMFSR and BPMFSRIC in details.

4.1. Bayesian Probabilistic Matrix Factorization with Social Relation

It is unreasonable that hyperparameters Θ_U are the same for different users in BPMF, which may cause some recommendation errors. To address this problem, we assume every user has its own hyperparameters. By this assumption, Eq. (5) should be modified as:

$$p(U) = \prod_{i=1}^{M} N\left(U_i|\mu_{U,i}, \Lambda_{U,i}^{-1}\right) \tag{11}$$

where $\Theta_{U,i} = \{\mu_{U,i}, \Lambda_{U,i}\}$ are the hyperparameters for user feature vector U_i.

The posterior distribution over user feature vector U_i described in Eq. (9) should also be modified. In fact, the posterior distribution over user feature vector U_i is conditioned on the item feature matrix V, the observed rating matrix R and its own hyperparameters $\Theta_{U,i} = \{\mu_{U,i}, \Lambda_{U,i}\}$ under our assumption. The posterior distribution over U_i is given as:

$$\begin{aligned} &p\left(U_i|R, V, \Theta_{U,i}, \alpha\right) = N\left(U_i|\mu_{U,i}^*, \left(\Lambda_{U,i}^*\right)^{-1}\right) \\ &\sim\prod_{j=1}^{N}\left[N\left(R_{ij}|U_i^T V_j, \alpha^{-1}\right)\right]^{I_{ij}} p\left(U_i|\mu_{U,i}, \Lambda_{U,i}\right) \end{aligned} \tag{12}$$

where $\Lambda_{U,i}^* = \Lambda_{U,i} + \alpha\sum_{j=1}^{N}\left(V_j^T V_j\right)^{I_{ij}}$ and $\mu_{U,i}^* = (\Lambda_i^*)^{-1}\left(\alpha\sum_{j=1}^{N}(V_j R_{ij})^{I_{ij}} + \mu_{U,i}\Lambda_{U,i}\right)$.

Because a user's preference is influenced by his (or her) friends, we suppose that the conditional distribution over user hyperparameters is conditioned on feature vectors of user's friends. By this assumption, Eq. (10) should be modified as:

$$\begin{aligned} p\left(\Theta_{U,i}|U,\Theta_0\right) &= p\left(\Theta_{U,i}|U_{F,i},\Theta_0\right) = p\left(\mu_{U,i},\Lambda_{U,i}|U_{F,i},\Theta_0\right) \\ &= N\left(\mu_{U,i}|\mu^*_{U,i},\left(\beta^*_{U,i}\Lambda_{U,i}\right)^{-1}\right)W\left(\Lambda_{U,i}|W^*_{U,i},v^*_{U,i}\right) \end{aligned} \tag{13}$$

$$\mu^*_{U,i} = \frac{\beta_0\mu_0 + M_i\overline{U}_{(i)}}{\beta_0+M_i}, \beta^*_{U,i} = \beta_0 + M_i, v^*_{U,i} = v_0 + M_i,$$

$$W^*_{U,i} = \left({W_0}^{-1} + M_i\overline{S}_{U,i} + \frac{\beta_0 M_i}{\beta_0 + M_i}\left(\mu_0 - \overline{U}_{(i)}\right)^T\left(\mu_0 - \overline{U}_{(i)}\right)\right)^{-1},$$

$$\overline{U}_{(i)} = \frac{1}{M_i}\sum\nolimits_{j\in F_i} U_j, \overline{S}_{U,i} = \frac{1}{M_i}\sum\nolimits_{j\in F_i}\left(U_j - \overline{U}_{(i)}\right)^T\left(U_j - \overline{U}_i\right), M_i = |F_i|$$

where $U_{F,i}$ is the matrix composed by feature vector of user i and feature vectors of user i's friends, F_i is the friend set of user i and himself and $|\cdot|$ denotes the size of a set.

The method described above is called Bayesian Probabilistic Matrix Factorization with Social Relations (BPMFSR), and the graph model is shown in Fig. 1(c). In this model, user feature vector U_i is generated according to its own hyperparameters $\Theta_{U,i}$ for each user. We also use Gibbs sampling algorithm to sample user feature vectors and item feature vectors, which is given in Algorithm 1. In this algorithm, we first generate user hyperparameters $\Theta_{U,i}$ for each user by Eq. (13) and then generate user feature vector U_i with user hyperparameters $\Theta_{U,i}$. If a user has very few friends, for an example, less than D, the hyperparameter estimation according to the feature vectors of his (or her) friends is meaningless. So in Algorithm 1, if user i has sufficient friends, hyperparameters are sampled according to the feature vectors of user i's friends, otherwise, hyperparameters are sampled according to the feature vectors of all users. Item hyperparameters and item feature vectors are generated by the same way in BPMF.

Algorithm 1. Gibbs sampling for Bayesian Probabilistic Matrix Factorization with Social Relationship (BPMFSR)

```
Initialize model parameters $U_i^{(1)}, V_j^{(1)}$;
for $t = 1...T$
  sample the hyperparameters(Eq.10):
    $\Theta_U^{(t)} \sim p(\Theta_U|U^{(t)},\Theta_0)$
    $\Theta_V^{(t)} \sim p(\Theta_V|V^{(t)},\Theta_0)$
  for $i = 1...M$,
    if $M_i < D$ then
      sample user features vector (Eq.9): $U_i^{(t+1)} \sim p(U_i|R,V^{(t)},\Theta_U^{(t)})$
    else
      sample hyperparameters for user $i$ (Eq.13): $\Theta_{U,i}^{(t)} \sim p(\Theta_{U,i}|U^{(t)},\Theta_0)$
      sample user feature vector (Eq.12):$U_i^{(t+1)} \sim p(U_i|R,V^{(t)},\Theta_{U,i}^{(t)})$
    end if
  end for
  for $j = 1...N$,
    sample item features vector: $V_j^{(t+1)} \sim p(V_j|R,U^{(t+1)},\Theta_V^{(t)})$
  end for
end for
```

Because different user hyperparameters are applied to different users, BPMFSR can get rid of the prediction errors caused by uniform user hyperparameters in BPMF. Furthermore, social relations are integrated in the model, which can improve prediction accuracy and alleviate the data sparsity problem and the cold-start problem.

4.2. Bayesian Probabilistic Matrix Factorization with Social Relation and Item Contents

Using uniform hyperparameters for all of the items in BPMF encounters the same problem caused by uniform user hyperparameters. To avoid the prediction error caused by uniform item hyperparameters and to further improve the recommendation accuracy, we assume that every item has its own hyperparameters. So Eq. (6) should be modified as:

$$p(V) = \prod_{j=1}^{N} N\left(V_j|\mu_{V,j},\Lambda_{V,j}^{-1}\right) \tag{14}$$

where $\Theta_{V,j} = \{\mu_{V,j},\Lambda_{V,j}\}$ are the hyperparameters for item feature vector V_j. By this assumption, the posterior distribution over item feature vector V_j is conditioned on the user feature matrix U, the observed rating matrix R and its own hyperparameters $\Theta_{V,j} = \{\mu_{V,j},\Lambda_{V,j}\}$. It is given as:

$$\begin{aligned} &p\left(V_j|R,U,\Theta_{V,j},\alpha\right) = N\left(V_j|\mu^*_{V,j},\left(\Lambda^*_{V,j}\right)^{-1}\right) \\ &\sim\prod_{i=1}^{M}\left[N\left(R_{ij}|U_i^T V_j,\alpha^{-1}\right)\right]^{I_{ij}} p\left(V_j|\mu_{V,j},\Lambda_{V,j}\right) \end{aligned} \tag{15}$$

where $\Lambda^*_{V,j} = \Lambda_{V,j} + \alpha\sum\nolimits_{i=1}^{M}(U_j^T U_j)^{I_{ij}}$ and $\mu^*_{V,j} = (\Lambda^*_{V,j})^{-1}\left(\alpha\sum\nolimits_{i=1}^{M}(U_i R_{ij})^{I_{ij}} + \mu_{V,j}\Lambda_{V,j}\right)$.

Let C_j denote the item set in which every item links to item j. The links between items can be constructed according to contents, such as item tags, categories and properties. For example, if two items are attached with a same tag, there is a link between them. We can also link the similar items by measuring item properties similarity.

To fuse item contents, we assume that the conditional distribution over item hyperparameters $\Theta_{V,j} = \{\mu_{V,j},\Lambda_{V,j}\}$ is only conditioned on the feature vectors of items in C_j. The underlying assumption is that the items with links should receive similar ratings. The conditional distribution over hyperparameters of item j, $\Theta_{V,j} = \{\mu_{V,j},\Lambda_{V,j}\}$, is given by:

$$\begin{aligned} p\left(\Theta_{V,j}|V,\Theta_0\right) &= p\left(\Theta_{V,j}|V_{C,j},\Theta_0\right) = p\left(\mu_{V,j},\Lambda_{V,j}|V_{C,j},\Theta_0\right) \\ &= N\left(\mu_{V,j}|\mu^*_{V,j},\left(\beta^*_{V,j}\Lambda_{V,j}\right)\right)W\left(\Lambda_{V,j}|W^*_{V,j},v^*_{V,j}\right) \end{aligned} \tag{16}$$

$$\mu^*_{V,j} = \frac{\beta_0\mu_0 + N_j\overline{V}_{(j)}}{\beta_0+N_j}, \beta^*_{V,j} = \beta_0 + N_j, v^*_{V,j} = v_0 + N_j,$$

$$W^*_{V,j} = \left({W_0}^{-1} + N_j\overline{S}_{V,j} + \frac{\beta_0 N_j}{\beta_0 + N_j}\left(\mu_0 - \overline{V}_{(j)}\right)^T\left(\mu_0 - \overline{V}_{(j)}\right)\right)^{-1},$$

$$\overline{V}_{(j)} = \frac{1}{N_j}\sum_{k\in C_j} V_k, \overline{S}_{V,j} = \frac{1}{N_j}\sum_{k\in C_j}\left(V_k - \overline{V}_{(j)}\right)^T\left(V_k - \overline{V}_{(j)}\right), N_j = \left|C_j\right|$$

where $V_{C,j}$ is the matrix composed by feature vectors of items in C_j, and C_j is the item set in which items are linked to item j.

Table 1
Statistics of datasets.

Dataset	Douban	Epinions	Last.fm
Num. of users	129,490	49,290	1892
Num. of items	58,541	139,783	17,632
Num. of ratings	16,830,839	664,824	92,834
Rating matrix density	2.220×10^{-3}	9.649×10^{-5}	2.783×10^{-3}
Num. of friend links	1,692,952	487,181	12,717
Num. of tag statements	0	0	186,479

This method is called Bayesian Probabilistic Matrix Factorization with Social Relations and Item Contents (BPMFSRIC). Graphical model for BPMFSRIC is shown in Fig. 1(d). In this model, item hyperparameters are generated for each item, as well as user hyperparameters. We also use Gibbs sampler to sample user feature vectors and item feature vectors, which is given in Algorithm 2. Similar to Algorithm 1, Algorithm 2 samples item hyperparameters for each item if there are more than D items linking to it.

Algorithm 2. Gibbs sampling for Bayesian Probabilistic Matrix Factorization with Social Relationship and Item Contents (BPMFSRIC)

```
Initialize model parameters $U_i^{(1)}, V_j^{(1)}$;
for $t = 1...T$
  sample the hyperparameters(Eq.10):
    $\Theta_U^{(t)} \sim p(\Theta_U|U^{(t)}, \Theta_0)$
    $\Theta_V^{(t)} \sim p(\Theta_V|V^{(t)}, \Theta_0)$
  for $i = 1...M$,
    if $M_i < D$ then
      sample user feature vector (Eq.9):$U_i^{(t+1)} \sim p(U_i|R, V^{(t)}, \Theta_U^{(t)})$
    else
      sample hyperparameters for user $i$ (Eq.13):$\Theta_{U,i}^{(t)} \sim p(\Theta_{U,i}|U^{(t)}, \Theta_0)$
      sample user feature vector (Eq.12): $U_i^{(t+1)} \sim p(U_i|R, V^{(t)}, \Theta_{U,i}^{(t)})$
    end if
  end for
  for $j = 1...N$,
    if $N_j < D$ then
      sample item feature vector: $V_j^{(t+1)} \sim p(V_j|R, U^{(t+1)}, \Theta_V^{(t)})$
    else
      sample hyperparameters for item $j$ (Eq.16):$\Theta_{V,j}^{(t)} \sim p(\Theta_{V,j}|V^{(t)}, \Theta_0)$
      sample item feature vector (Eq.15):$V_j^{(t+1)} \sim p(V_j|R, U^{(t+1)}, \Theta_{V,j}^{(t)})$
    end if
  end for
end for
```

4.3. Complexity analysis

The main computation of Algorithms 1 and 2 is sampling user feature vectors using Eq. (12) and sampling item feature vectors using Eq. (15). The computational complexity of sampling all user feature vectors and sampling all item feature vectors is $O(K)$, where K is the number of nonzero entries in rating matrix R. So the computational complexity of one iteration in Algorithms 1 and 2 is $O(K)$, which indicates that the computational complexity of our method is linear with respect to the number of observed ratings. This complexity analysis shows that our methods are very efficient and can scale up with respect to very large datasets.

Compared with BPMF, the computational complexity of our methods is slightly higher in one iteration, because Algorithms 1 and 2 sample user hyperparameters and item hyperparameters using Eqs. (13) and (16) additionally. However, the computational complexity of sampling all user hyperparameters and sampling all item hyperparameters is much less than that of sampling user feature vectors and sampling item feature vectors. Furthermore, the experiments in Section 5 show that our methods converge faster than BPMF. Considering both computational complexity in one iteration and converge speed, our methods can achieve the same recommendation accuracy in few iterations and in less time.

4.4. Convergence analysis

Because we infer our model through Gibbs sampling, the convergence of the proposed method is guaranteed by that of Gibbs sampler. It has been proved that under positivity conditions, the Markov chain generated by Gibbs sampler will converge to its invariant distribution.

Definition 1. Positivity condition [21]

A density function $f(x_1,x_2,\ldots,x_n)$ and marginal density functions $f_i(x_i)$ are said to satisfy the positivity condition if $f_i(x_i) > 0$ for all $x_1, x_2,\ldots, x_n$ implies that $f(x_1,x_2,\ldots,x_n) > 0$.

Lemma 1. [21] *If the joint distribution $f(x_1,x_2,\ldots,x_n)$ satisfies the positivity condition, the Gibbs sampler yields an irreducible and recurrent Markov chain.*

The distribution in our model satisfies the positivity condition, so our methods converge as the number of iteration approaches to ∞. Convergence of the proposed method is also confirmed by the experiments in Section 5.

5. Experiments

In this section, we present the experimental results on three large scale datasets to compare our method with other recommendation methods based on matrix factorization. We also verify our methods in cold-start settings.

Table 2
Predictive accuracy comparison on Douban Dataset (the values in brackets are the results reported in [16]).

D (dimension)	Training	Metrics	Hao Ma's method	BPMF	BPMFSR
10	40%	MAE	0.5845 ± 0.0008(0.5685)	0.5583 ± 0.0001	**0.5577 ± 0.0001**
		RMSE	0.7411 ± 0.0005(0.7125)	0.7045 ± 0.0002	0.7045 ± 0.0001
	60%	MAE	0.5665 ± 0.0005(0.5593)	0.5512 ± 0.0001	**0.5510 ± 0.0001**
		RMSE	0.7221 ± 0.0005(0.7042)	**0.6971 ± 0.0001**	0.6973 ± 0.0001
	80%	MAE	0.5562 ± 0.0003(0.5543)	0.5472 ± 0.0002	**0.5467 ± 0.0002**
		RMSE	0.7099 ± 0.0004(0.6988)	**0.6930 ± 0.0002**	0.6934 ± 0.0003
30	40%	MAE	0.5950 ± 0.0004	0.5588 ± 0.0001	**0.5570 ± 0.0001**
		RMSE	0.7524 ± 0.0005	0.7051 ± 0.0001	**0.7039 ± 0.0001**
	60%	MAE	0.5789 ± 0.0002	0.5509 ± 0.0001	**0.5486 ± 0.0001**
		RMSE	0.7336 ± 0.0007	0.6963 ± 0.0001	**0.6948 ± 0.0001**
	80%	MAE	0.5674 ± 0.0001	0.5448 ± 0.0001	**0.5427 ± 0.0001**
		RMSE	0.7204 ± 0.0001	0.6896 ± 0.0001	**0.6884 ± 0.0001**

Please cite this article as: J. Liu, et al., Bayesian Probabilistic Matrix Factorization with Social Relations and Item Contents for recommendation, Decision Support Systems (2013), http://dx.doi.org/10.1016/j.dss.2013.04.002

Table 3
Predictive accuracy comparison on Epinions dataset.

D (dimension)	Training	Metrics	Hao Ma's method	BPMF	BPMFSR
10	40%	MAE	0.9261 ± 0.0034	0.8535 ± 0.0018	**0.8411 ± 0.0006**
		RMSE	1.2046 ± 0.0050	1.0858 ± 0.0026	**1.0695 ± 0.0008**
	60%	MAE	0.9313 ± 0.0023	0.8383 ± 0.0018	**0.8359 ± 0.0006**
		RMSE	1.2003 ± 0.0047	1.0704 ± 0.0027	**1.0655 ± 0.0009**
	80%	MAE	0.9018 ± 0.0026	0.8144 ± 0.0020	**0.8114 ± 0.0022**
		RMSE	1.1630 ± 0.0027	1.0498 ± 0.0023	**1.0435 ± 0.0025**
	90%	MAE	0.8915 ± 0.0036	0.8081 ± 0.0020	**0.8056 ± 0.0020**
		RMSE	1.1510 ± 0.0055	1.0435 ± 0.0035	**1.0334 ± 0.0040**
30	40%	MAE	0.9341 ± 0.0040	0.8446 ± 0.0073	**0.8381 ± 0.0010**
		RMSE	1.2043 ± 0.0021	1.0785 ± 0.0069	**1.0666 ± 0.0012**
	60%	MAE	0.9332 ± 0.0031	0.8423 ± 0.0057	**0.8348 ± 0.0017**
		RMSE	1.2015 ± 0.0039	1.0741 ± 0.0056	**1.0640 ± 0.0025**
	80%	MAE	0.9135 ± 0.0019	0.8156 ± 0.0018	**0.8124 ± 0.0016**
		RMSE	1.1736 ± 0.0025	1.0496 ± 0.0026	**1.0403 ± 0.0024**
	90%	MAE	0.9078 ± 0.0039	0.8087 ± 0.0025	0.8078 ± 0.0024
		RMSE	1.1661 ± 0.0051	1.0434 ± 0.0034	**1.0352 ± 0.0030**

5.1. Datasets

We evaluate our method on Douban dataset [16], Epinions dataset [17] and Last.fm dataset [4]. Douban (http://www.douban.com) dataset, crawled by Hao Ma et al. [16], contains 16,830,839 ratings of 129,490 users on 58,541 movies and 1,692,952 friend links between these users. For more details, please see [16].

Epinions (http://www.epinions.com) dataset [17] contains 49,290 users and 139,783 items. These users issued 664,824 ratings and 487,181 trust statements. Note that the Epinions dataset collected by Massa and Avesani [17] used in this paper is different from that used in [16]. Epinions dataset collected by Massa and Avesani [17] contains more items, ratings, friend links and fewer users.

Last.fm (http://www.lastfm.com) dataset is released in the framework of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011) [4]. It contains 1892 users, 17,632 artists and 11,946 tags. Different from Douban dataset and Epinions dataset, Last.fm dataset only records listening count that each user listened to artists. Listening count ranges from 1 to 352,698. To test our methods on Last.fm, we map listening counts into integer values of 1 to 5 to represent the extent of favor of artists by the similar way in [7]. The mapping formula is given as:

$$r = \begin{cases} \lfloor \log_{10} l \rfloor + 1, \text{if} \lfloor \log_{10} l \rfloor + 1 \le 5 \\ 5, \text{otherwise} \end{cases} \tag{17}$$

where *l* is the listening count, *r* is the mapped value, and ⌊·⌋ is the operator of rounding towards zero. To test BPMFSRIC method, we build the links according to tags. If two artists received a same tag more than 5 times,

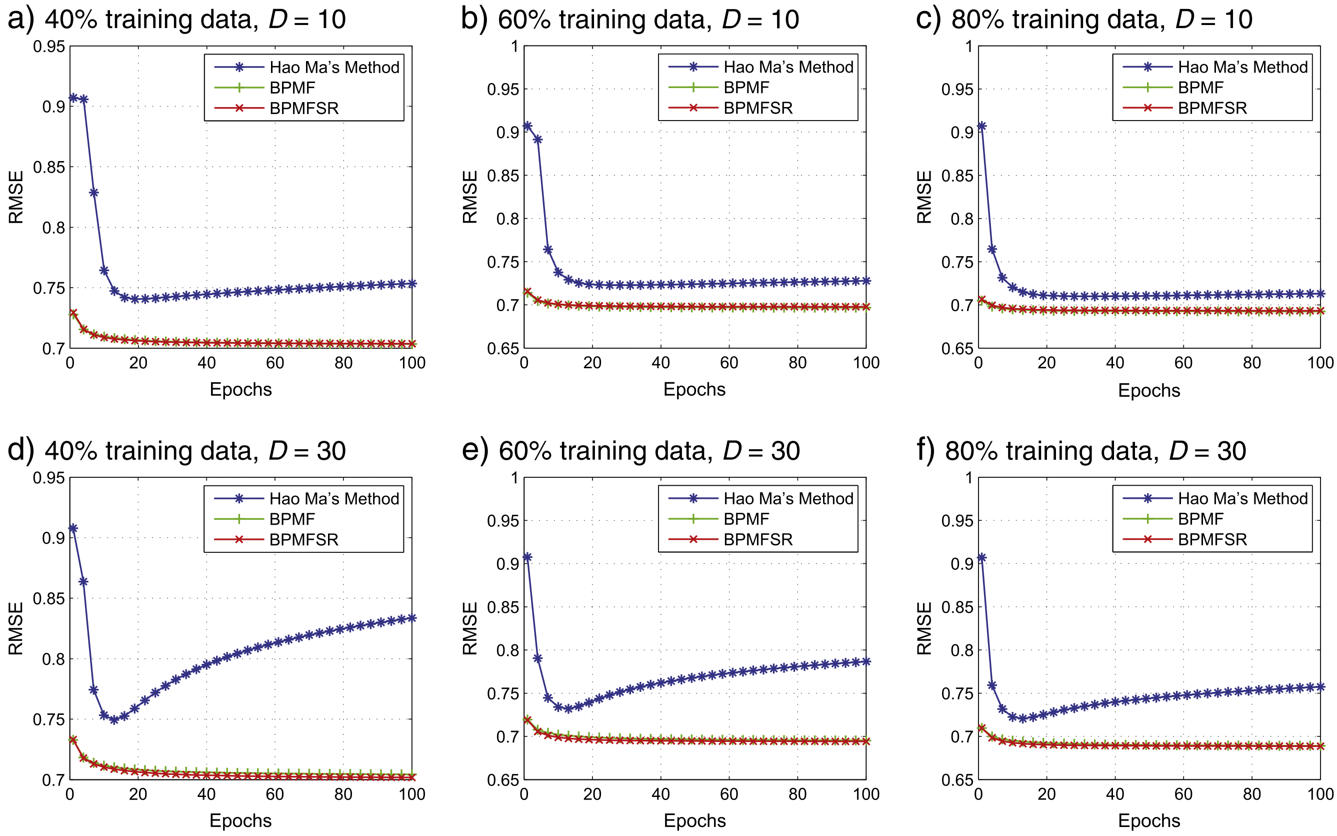


Fig. 2. Testing RMSE of all methods on Douban dataset, the x-axis shows the number of epoch and y-axis shows testing RMSE.

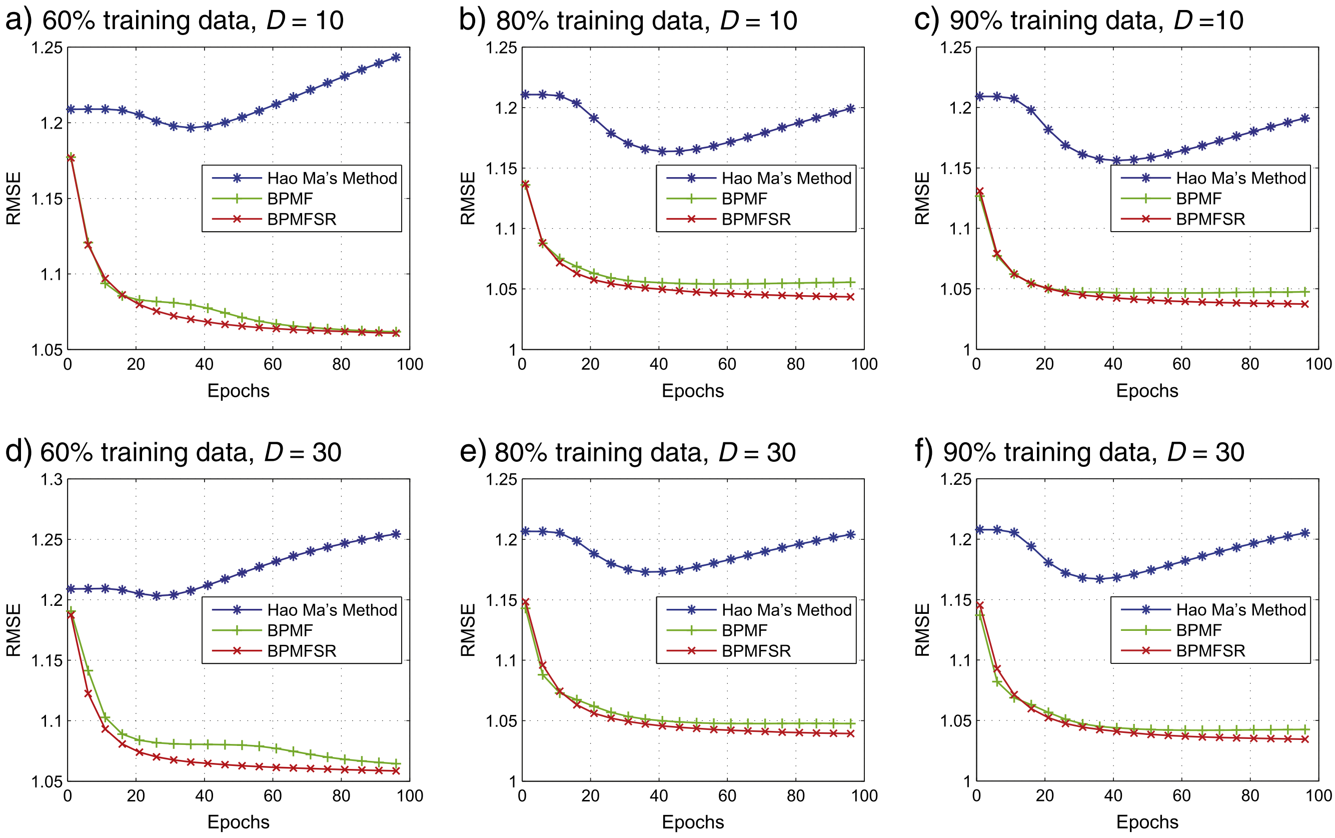


Fig. 3. Testing RMSE of all methods on Epinions dataset, the x-axis shows the number of epoch and y-axis shows testing RMSE.

we link these two artists. Last.fm contains 12,717 friend links, 92,834 listening counts and 186,479 tag statements.

The statistics of these datasets are summarized in Table 1.

We use Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to measure prediction accuracy of recommendation methods. MAE is defined as:

$$\mathrm{MAE} = \frac{1}{T}\sum_{i,j}\left|R_{ij} - \hat{R}_{ij}\right| \tag{18}$$

where R_{ij} is the rating given by user i for item j, and $\hat{R}_{ij}$ is the prediction of R_{ij}. T is the total number of tested ratings. RMSE is defined as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum_{i,j}\left(R_{ij} - \hat{R}_{ij}\right)^2}. \tag{19}$$

5.2. Comparisons

We compare Hao Ma's method [16] and BPMF [23] with our methods. Hao Ma's method is implemented using model 2 with PCC similarity (see [16] for details). It is reported that this implementation gets the highest prediction accuracy and outperforms NMF [8], PMF [22], RSTE [14] and other state-of-art. The initial solutions of our methods are the same as those of BPMF. In our methods, μ_0, v_0, and W_0 are set to be the same values as those in BPMF. The experiments repeat 10 times. Mean and standard deviation of MAE and RMSE are calculated.

For Douban dataset, we randomly select 40%, 60% and 80% ratings as training data, and use the rest of ratings to test the algorithms. For Epinions dataset, we use 40%, 60%, 80% and 90% ratings as training data. Because Douban dataset and Epinions dataset don't contain item contents information, which is needed by BPMFSRIC, we don't test BPMFSRIC on these two datasets. The experimental results are shown in Tables 2 and 3. We give the mean and standard deviation of RMSE

Table 4
Predictive accuracy comparison on Last.fm dataset.

D (dimension)	Training	Metrics	Hao Ma's method	BPMF	BPMFSR	BPMFSRIC
10	40%	MAE	0.4806 ± 0.0032	0.3359 ± 0.0018	**0.3338 ± 0.0012**	0.3341 ± 0.0007
		RMSE	0.6936 ± 0.0047	0.4655 ± 0.0036	0.4613 ± 0.0028	**0.4604 ± 0.0026**
	60%	MAE	0.4539 ± 0.0070	0.3270 ± 0.0014	**0.3261 ± 0.0015**	0.3278 ± 0.0012
		RMSE	0.6529 ± 0.0104	0.4502 ± 0.0023	**0.4489 ± 0.0023**	0.4502 ± 0.0021
	80%	MAE	0.4310 ± 0.0056	0.3234 ± 0.0012	0.3222 ± 0.0014	0.3237 ± 0.0016
		RMSE	0.6210 ± 0.0080	0.4465 ± 0.0018	0.4449 ± 0.0021	0.4461 ± 0.0020
30	40%	MAE	0.4849 ± 0.0020	0.3378 ± 0.0006	0.3354 ± 0.0004	0.3345 ± 0.0005
		RMSE	0.6977 ± 0.0019	0.4686 ± 0.0018	0.4630 ± 0.0017	**0.4596 ± 0.0013**
	60%	MAE	0.4691 ± 0.0027	0.3283 ±0.0016	0.3277 ± 0.0019	0.3286 ± 0.0017
		RMSE	0.6708 ± 0.0038	0.4529 ± 0.0021	0.4515 ± 0.0022	0.4508 ± 0.0020
	80%	MAE	0.4492 ± 0.0013	0.3241 ± 0.0012	0.3235 ± 0.0012	0.3244 ± 0.0014
		RMSE	0.6418 ± 0.0026	0.4467 ± 0.0024	0.4452 ± 0.0024	0.4451 ± 0.0026

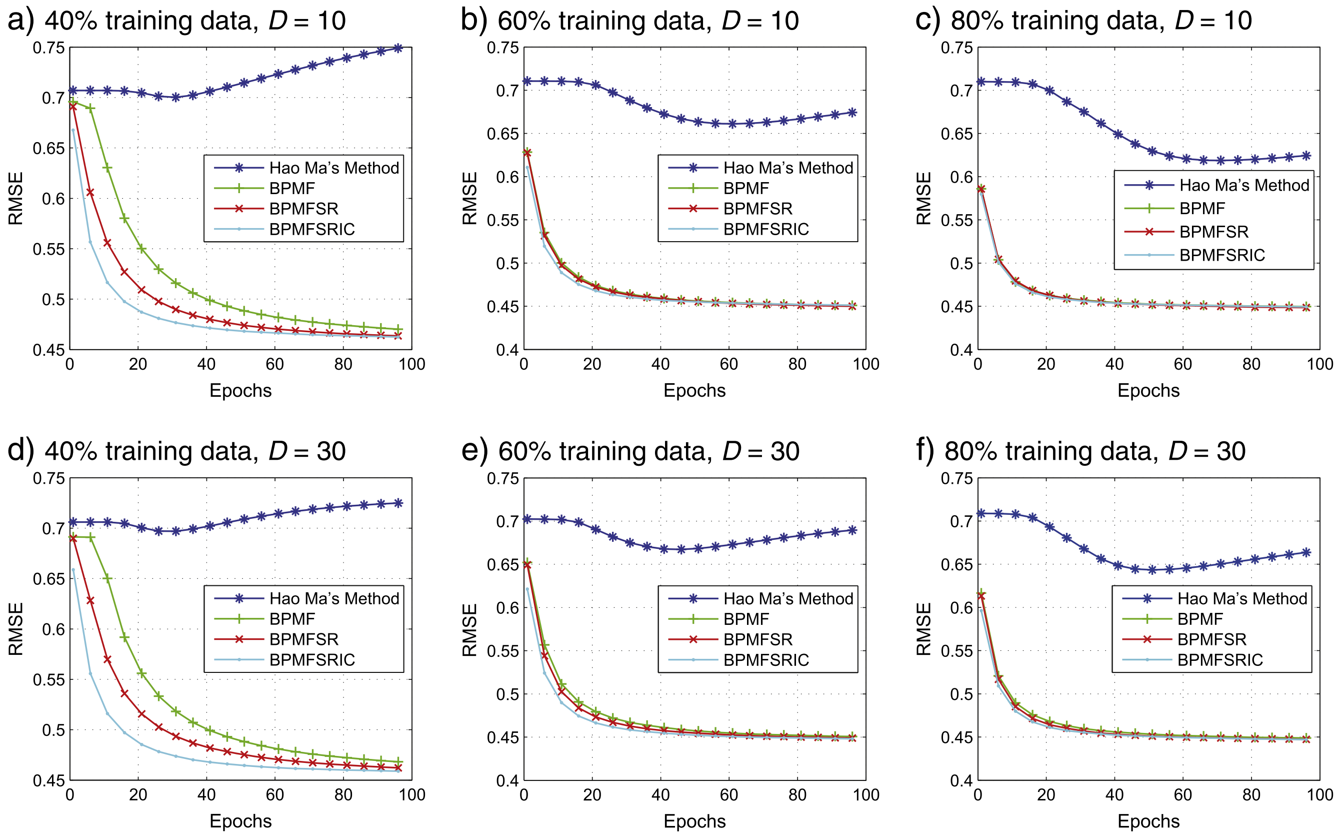


Fig. 4. Testing RMSE of all methods on Last.fm dataset, the x-axis shows the number of epoch and y-axis shows testing RMSE.

and MAE for each experiment. The best results which are statistically significant (at the 5% significance level) are set to be bold. According to the results, it can be observed that our method outperforms Hao Ma's method and BPMF on Epinions datasets in the term of improved RMSE and MAE. On Douban dataset, in the cases of 60% and 80% training data with $D = 10$, BPMF get lower RMSE than BPMFSR. MAE and RMSE

Table 5
Predictive accuracy comparison in cold-start settings on Douban dataset.

D (dimension)	Cold-start users	Metrics	Hao Ma's method	BPMF	BPMFSR
10	60%	MAE	0.7321 ± 0.0005	0.6337 ± 0.0004	0.6335 ± 0.0003
		RMSE	0.9067 ± 0.0002	0.7902 ± 0.0004	**0.7885 ± 0.0004**
	40%	MAE	0.7319 ± 0.0003	0.6333 ± 0.0004	**0.6327 ± 0.0003**
		RMSE	0.9071 ± 0.0006	0.7899 ± 0.0004	**0.7877 ± 0.0005**
30	60%	MAE	0.7320 ± 0.0008	0.6340 ± 0.0007	**0.6336 ± 0.0009**
		RMSE	0.9077 ± 0.0010	0.7906 ± 0.0010	**0.7895 ± 0.0012**
	40%	MAE	0.7324 ± 0.0007	0.6338 ± 0.0006	**0.6334 ± 0.0007**
		RMSE	0.9080 ± 0.0010	0.7903 ± 0.0011	**0.7891 ± 0.0012**

Table 6
Predictive accuracy comparison in cold-start settings on Epinions dataset.

D (dimension)	Cold-start users	Metrics	Hao Ma's method	BPMF	BPMFSR
10	60%	MAE	0.9236 ± 0.0026	0.8558 ± 0.0021	0.8558 ± 0.0017
		RMSE	1.2092 ± 0.0031	1.0809 ± 0.0034	1.0806 ± 0.0033
	40%	MAE	0.9250 ± 0.0008	0.8517 ± 0.0012	0.8517 ± 0.0010
		RMSE	1.2105 ± 0.0015	1.0769 ± 0.0018	1.0759 ± 0.0018
	10%	MAE	0.9229 ± 0.0025	0.8496 ± 0.0040	0.8469 ± 0.0049
		RMSE	1.2104 ± 0.0040	1.0725 ± 0.0053	1.0717 ± 0.0055
	5%	MAE	0.9192 ± 0.0049	0.8424 ± 0.0050	0.8389 ± 0.0048
		RMSE	1.2071 ± 0.0048	1.0643 ± 0.0055	1.0625 ± 0.0049
30	60%	MAE	0.9287 ± 0.0030	0.8551 ± 0.0025	**0.8516 ± 0.0024**
		RMSE	1.2093 ± 0.0031	1.0803 ± 0.0030	1.0794 ± 0.0029
	40%	MAE	0.9264 ± 0.0014	0.8478 ± 0.0013	0.8469 ± 0.0011
		RMSE	1.2064 ± 0.0023	1.0740 ± 0.0012	1.0742 ± 0.0013
	10%	MAE	0.9282 ± 0.0048	0.8498 ± 0.0040	0.8449 ± 0.0039
		RMSE	1.2113 ± 0.0066	1.0730 ± 0.0054	1.0702 ± 0.0052
	5%	MAE	0.9243 ± 0.0108	0.8431 ± 0.0063	0.8388 ± 0.0071
		RMSE	1.2082 ± 0.0152	1.0655 ± 0.0090	1.0623 ± 0.0101

Please cite this article as: J. Liu, et al., Bayesian Probabilistic Matrix Factorization with Social Relations and Item Contents for recommendation, Decision Support Systems (2013), http://dx.doi.org/10.1016/j.dss.2013.04.002

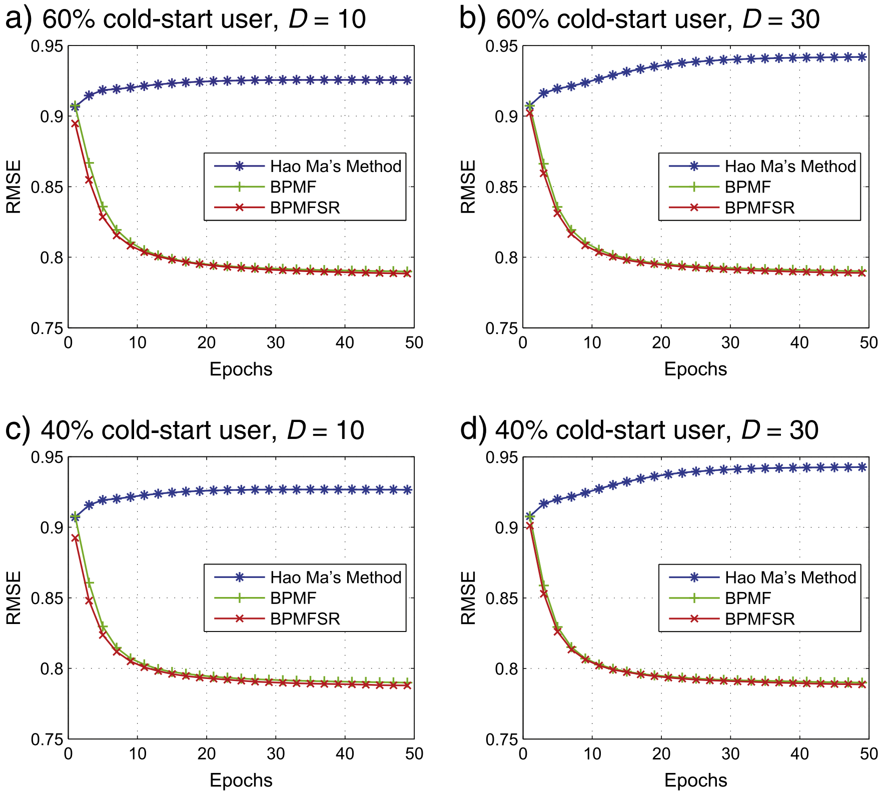


Fig. 5. Testing RMSE of all methods on Douban dataset for cold-start user setting, the x-axis shows the number of iteration and y-axis shows testing RMSE.

generated by BPMF and BPMFSR on Douban dataset are very close. The advantages of BPMFSR are more obvious on Epinions dataset. Epinions dataset is much sparser than Douban dataset, so we can say that BPMFSR alleviates the data sparsity problem better than other methods. We also notice that the increasing of user and item feature vector dimension doesn't improve the predictive accuracy of Hao Ma's method

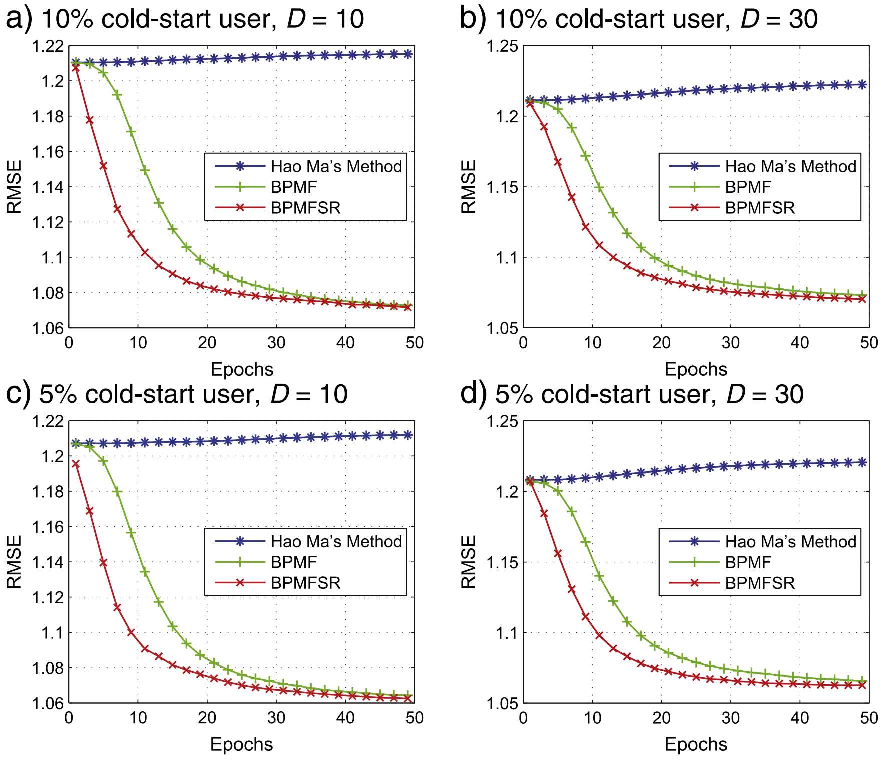


Fig. 6. Testing RMSE of all methods on Epinions dataset for cold-start user setting, the x-axis shows the number of epoch and y-axis shows testing RMSE.

Table 7
Predictive accuracy comparison in cold-start settings on Last.fm dataset.

D (dimension)	Cold-start users/items	Metrics	Hao Ma's method	BPMF	BPMFSR	BPMFSRIC
10	10%/10%	MAE	0.4973 ± 0.0155	0.4213 ± 0.0183	0.4209 ± 0.0186	0.4178 ± 0.0169
		RMSE	0.7150 ± 0.0180	0.6039 ± 0.0225	0.6009 ± 0.0239	0.5960 ± 0.0191
	5%/5%	MAE	0.4860 ± 0.0187	0.4127 ± 0.0183	0.4106 ± 0.0183	0.4082 ± 0.0198
		RMSE	0.7027 ± 0.0232	0.5945 ± 0.0260	0.5885 ± 0.0276	0.5857 ± 0.0271
30	10%/10%	MAE	0.4883 ± 0.0095	0.4174 ± 0.0105	0.4143 ± 0.0086	**0.4131** ± **0.0087**
		RMSE	0.7091 ± 0.0101	0.6017 ± 0.0114	0.5972 ± 0.0100	0.5954 ± 0.0097
	5%/5%	MAE	0.4800 ± 0.0183	0.4093 ± 0.0165	0.4055 ± 0.0159	0.4033 ± 0.0160
		RMSE	0.6970 ± 0.0211	0.5878 ± 0.0212	0.5831 ± 0.0199	0.5786 ± 0.0203

and BPMF, while BPMFSR gets more accurate results with higher dimension. This indicates that Hao Ma's method and BPMF may over fit in high dimension.

Note that the results reported in [16], shown in the brackets in Table 2, are slightly lower than the results obtained by our implementation. Because we cannot get the code used in [16], we attribute the discrepancy to some implementation details unmentioned in [16]. Although the results reported in [16] and those obtained by us are slightly different, we can draw the same conclusion that our method outperforms Hao Ma's method on Douban dataset in the case of $D = 10$ even using the results reported in [16].

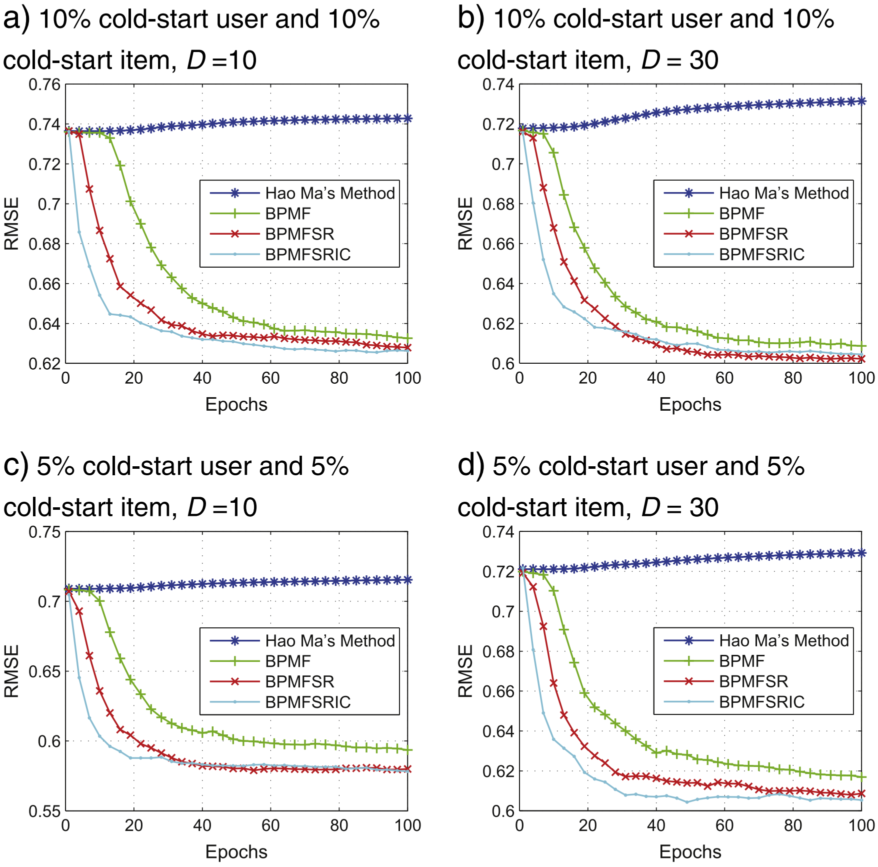
Figs. 2 and 3 show testing RMSE generated by all methods at every epoch on Douban dataset and Epinions dataset. It can be observed that Hao Ma's method over fits after several epochs, while BPMF and BPMFSR do not over fit at all. RMSE values generated by BPMFSR are higher than those generated by BPMF at the beginning, but after about 10 epochs BPMFSR outperforms BPMF. This indicates that BPMFSR converges faster than BPMF.

We compare BPMFSR and BPMFSRIC with Hao Ma's method and BPMF on Last.fm dataset. Listening count is mapped into 5-point rating by Eq. (17). Tag statements are used to build links between items (artists). If two artists received a same tag more than 5 times, we link these two artists. We randomly select 40%, 60% and 80% ratings as training data, and use the rest of ratings as testing data. The experiments repeat 10 times, and mean and standard deviation of MAE and RMSE are calculated. The experimental results are shown in Table 4. It can be observed that our methods get lower mean of MAE and RMSE than Hao Ma's method and BPMF in all cases. When dimension is high, BPMFSRIC gets lower mean of MAE and RMSE than BPMFSR.

Fig. 4 shows testing RMSE generated by all methods at every epoch on Last.fm dataset. Trends of convergence are similar to those in Figs. 2 and 3. Furthermore, BPMFSRIC and BPMFSR converge faster than BPMF especially in the cases of 40% training data.

5.3. Comparisons in cold-start settings

We test our methods in cold-start settings. For Douban dataset we randomly select 60% and 40% users as cold-start users. For Epinions dataset, we randomly select 60%, 40%, 10% and 5% users as cold-start users. All the ratings stated by cold-start users are treated as testing


a) 10% cold-start user and 10% cold-start item, *D* =10

b) 10% cold-start user and 10% cold-start item, *D* = 30

c) 5% cold-start user and 5% cold-start item, *D* =10

d) 5% cold-start user and 5% cold-start item, *D* = 30

Fig. 7. Testing RMSE of all methods on Last.fm dataset for cold-start user setting, the x-axis shows the number of epoch and y-axis shows testing RMSE.

Please cite this article as: J. Liu, et al., Bayesian Probabilistic Matrix Factorization with Social Relations and Item Contents for recommendation, Decision Support Systems (2013), http://dx.doi.org/10.1016/j.dss.2013.04.002

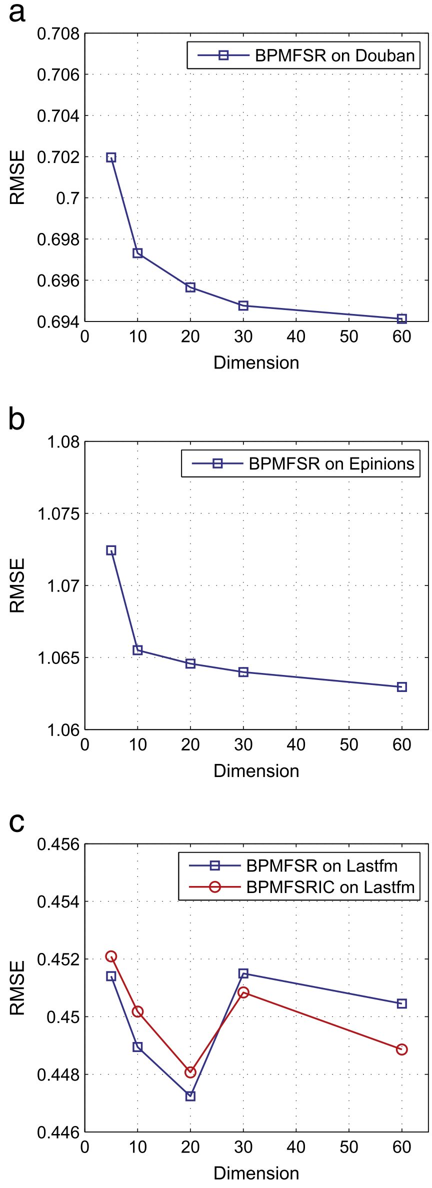


Fig. 8. Impact of feature dimension on all datasets (60% training data).

data and all the other ratings are treated as training data. We compare Hao Ma's method and BPMF with BPMFSR. The results are shown in Tables 5 and 6. It can be observed that BPMFSR outperforms the compared methods on Douban dataset in all cases. In Epinions dataset, BPMFSR obtains lower mean of RMSE and MAE.

Figs. 5 and 6 show testing RMSE generated by all methods at every epoch on Douban dataset and Epinions dataset in cold-start-user settings. We can find Hao Ma's method overfits at the beginning in all cases. BPMFSR converges faster than BPMF and gets better results in fewer epochs.

For Last.fm dataset we randomly select 5% and 10% users and items as cold-start users and cold-start items. All the ratings for cold-start users and cold-start items are treated as testing data and all other ratings are treated as training data. We compare Hao Ma's method and BPMF with BPMFSR and BPMFSRIC. The results are shown in Table 7. It can be observed that BPMFSR and BPMFSRIC obtain lower mean of RMSE and MAE then the compared methods in all cases. Because item contents are used in BPMFSRIC, BPMFSRIC gets more accurate prediction results than BPMFSR in all cases.

Fig. 7 shows the testing RMSE generated by all methods at every epoch on Last.fm dataset in cold-start user and item settings. We can find Hao Ma's method overfits at the beginning for all cases. Our methods converge faster than BPMF and get better results in fewer epochs.

5.4. Impact of the feature dimension

We investigate the impact of the feature dimension. Using 60% training data, we change the feature dimension and calculate RMSE on Douban dataset, Epinions dataset and Last.fm dataset. The results are presented in Fig. 8. We observe that the feature dimension impacts the recommendation results. As feature dimension increases, the prediction accuracy increases quickly at first. But when feature dimension increases further, the prediction accuracy increases slowly and even decreases on Last.fm. This phenomenon indicates that very high feature dimension cannot help to improve the recommendation accuracy.

6. Conclusion and future work

To address the data sparsity problem and the cold-start problem, in this paper, we modify the model in BPMF. We assume that the user hyperparameters and item hyperparameters are different for each user vector and item vector. The proposed recommendation methods, BPMFSR and BPMFSRIC, sample user hyperparameters and item hyperparameters according to the social relations and item contents. By this novel way we fuse social relations and item contents with ratings, which is different from traditional regularization-based methods and factorization-based methods. BPMFSR can be applied to trust-aware recommendation systems, while if the item contents are available, BPMFSRIC can improve recommendation accuracy further. The proposed methods are computationally efficient and can scale up with respect to very large datasets. Experimental results on three large real world datasets show that our methods get more accurate recommendation results with faster converging speed than the other state-of-the-art recommendation methods based on matrix factorization. Moreover, our methods outperform other methods in cold-start settings.

In our methods, we only use the trust information, while distrust statements are also provided in many online social networks. How to use distrust information is one of our further research directions. An important problem should be investigated: how the distrust relations affect the user preference.

Furthermore, we only use the direct trust relations between users and ignore the indirect trust relations. Trust relations can propagate among people in real life, so indirect trust relations can also affect the preference of a user. How to fuse indirect trust relations is another research direction.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61173120). The authors would like to thank the reviewers and editor for their helpful comments.

References

[1] R.P. Adams, G.E. Dahl, I. Murray, Incorporating Side Information in Probabilistic Matrix Factorization with Gaussian Processes, UAI, 2010, pp. 1–9.

[2] G. Adomavicius, 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 17 (6) (2005) 734–749.

[3] S. Burer, R. Monteiro, A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization, Mathematical Programming 95 (2) (2003) 329–357.

[4] I. Cantador, P. Brusilovsky, T. Kuflik, Second workshop on information heterogeneity and fusion in recommender systems (hetrec2011), Proceedings of the Fifth ACM Conference on Recommender systems, RecSys'11, ACM, New York, NY, USA, 2011, pp. 387–388.

[5] R. Gemulla, E. Nijkamp, P. Haas, Y. Sismanis, Large-scale matrix factorization with distributed stochastic gradient descent, Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2011, pp. 69–77.

[6] M. Jamali, M. Ester, A transitivity aware matrix factorization model for recommendation in social networks, Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, IJCAI'11, AAAI Press, 2011, pp. 2644–2649.

[7] O. Koyejo, J. Ghosh, A kernel-based approach to exploiting interaction-networks in heterogeneous information sources for improved recommender systems, Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems, ACM, 2011, pp. 9–16.

[8] D. Lee, H. Seung, et al., Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788–791.

[9] W. Li, D. Yeung, Relation regularized matrix factorization, Proceedings of the 21st International Joint Conference on, Artificial Intelligence, 2009, pp. 1126–1131.

[10] Y. Lim, Y. Teh, Variational Bayesian approach to movie rating prediction, Proceedings of KDD Cup and Workshop, Citeseer, 2007, pp. 15–21.

[11] Z. Lu, D. Agarwal, I. Dhillon, A spatio-temporal approach to collaborative filtering, Proceedings of the Third ACM Conference on Recommender Systems, ACM, 2009, pp. 13–20.

[12] X. Luo, Y. Xia, Q. Zhu, Incremental collaborative filtering recommender based on regularized matrix factorization, Knowledge-Based Systems 27 (2012) 271–280.

[13] H. Ma, H. Yang, M. Lyu, I. King, SoRec: social recommendation using probabilistic matrix factorization, Proceedings of the 17th ACM Conference on Information and Knowledge Management, ACM, 2008, pp. 931–940.

[14] H. Ma, I. King, M. Lyu, Learning to recommend with social trust ensemble, Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2009, pp. 203–210.

[15] H. Ma, T. Zhou, M. Lyu, I. King, Improving recommender systems by incorporating social contextual information, ACM Transactions on Information Systems (TOIS) 29 (2) (2011) 9:1–9:23.

[16] H. Ma, D. Zhou, C. Liu, M. Lyu, I. King, Recommender systems with social regularization, Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, ACM, 2011, pp. 287–296.

[17] P. Massa, P. Avesani, Trust-aware bootstrapping of recommender systems, ECAI 2006 Workshop on Recommender Systems, Citeseer, Riva del Garda, Italy, 2006, pp. 29–33.

[18] S. Nakajima, M. Sugiyama, Theoretical analysis of Bayesian matrix factorization, Journal of Machine Learning Research 12 (2011) 2583–2648.

[19] D.H. Park, H.K. Kim, I.Y. Choi, J.K. Kim, A literature review and classification of recommender systems research, Expert Systems with Applications 39 (11) (2012) 10059–10072.

[20] I. Porteous, A. Asuncion, M. Welling, Bayesian matrix factorization with side information and Dirichlet process mixtures, Proceedings of the 24th AAAI Conference on, Artificial Intelligence, 2010, pp. 563–568.

[21] C. Robert, G. Casella, Monte Carlo Statistical Methods, Springer Texts in Statistics, Springer, 2004.

[22] R. Salakhutdinov, A. Mnih, Probabilistic Matrix Factorization, Advances in Neural Information Processing Systems, 20, 2008. 1257–1264.

[23] R. Salakhutdinov, A. Mnih, Bayesian probabilistic matrix factorization using Markov Chain Monte Carlo, Proceedings of the 25th International Conference on Machine Learning, ACM, 2008, pp. 880–887.

[24] B. Sarwar, G. Karypis, J. Konstan, J. Reidl, Item-based collaborative filtering recommendation algorithms, Proceedings of the 10th International Conference on World Wide Web, ACM, 2001, pp. 285–295.

[25] H. Shan, A. Banerjee, Generalized probabilistic matrix factorizations for collaborative filtering, Data Mining (ICDM), 2010 IEEE 10th International Conference on, IEEE, 2010, pp. 1025–1030.

[26] A. Singh, G. Gordon, A Bayesian matrix factorization model for relational data, Proceedings of the Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-10), AUAI Press, Corvallis, Oregon, 2010, pp. 556–563.

[27] Q. Yuan, L. Chen, S. Zhao, Factorization vs. regularization: Fusing heterogeneous social relationships in top-n recommendation, Proceedings of the fifth ACM conference on Recommender Systems, ACM, 2011, pp. 245–252.

[28] J. Zhu, H. Ma, C. Chen, J. Bu, Social Recommendation Using Low-rank Semidefinite Program, AAAI, 2011, pp. 158–163.

Juntao Liu received the BS and MS degrees in Computer Science from Ordnance Engineering College, Shijiazhuang, China, in 2002 and 2005, respectively. He is a lecturer in the Department of Computer Engineering, Ordnance Engineering College. He is currently pursuing the Ph.D. degree in the Department of Electronics and Information Engineering, Huazhong University of Science and Technology. His research interests include data mining, machine learning and computer vision.

Caihua Wu received the BS, MS and PhD degrees in Computer Science from Ordnance Engineering College, Shijiazhuang, China, in 2003, 2006 and 2009 respectively. Now she is a lecturer in the Department of Information Counterwork, Air Force Radar Academy. Her research interests include data mining, information counterwork and software engineering.

Wenyu Liu received the BS degree in Computer Science from Tsinghua University, Beijing, China, in 1986, and the MS and PhD degrees, both in Electronics and Information Engineering, from Huazhong University of Science and Technology (HUST), Wuhan, China, in 1991 and 2001, respectively. He is now a professor and associate dean of the Department of Electronics and Information Engineering, HUST. His current research areas include computer graphics, multimedia information processing, and computer vision. He is a member of IEEE System, Man and Cybernetics Society.

Please cite this article as: J. Liu, et al., Bayesian Probabilistic Matrix Factorization with Social Relations and Item Contents for recommendation, Decision Support Systems (2013), http://dx.doi.org/10.1016/j.dss.2013.04.002