Modeling Implicit Communities in Recommender Systems

Lin Xiao[1(✉)] and Gu Zhaoquan[2]

[1] Institute of Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
jackielinxiao@gmail.com
[2] Department of Computer Science, GuangZhou Univeristy and The University of HongKong, Hongkong, China
demin456@gmail.com

Abstract. In recommender systems, a group of users may have similar preferences on a set of items. As the groups of users and items are not explicitly given, these similar-preferences groups are called implicit communities (where users inside same communities may not necessarily know each other).

Implicit communities can be detected with users' rating behaviors. In this paper, we propose a unified model to discover the implicit communities with rating behaviors from recommender systems.

Following the spirit of Latent Factor Model, we design a bayesian probabilistic graphical model which generates the implicit communities, where the latent vectors of users/items inside the same community follow the same distribution. An implicit community model is proposed based on rating behaviors and a Gibbs Sampling based algorithm is proposed for corresponding parameter inferences. To the best of our knowledge, this is the first attempt to integrate the rating information into implicit communities for recommendation.

We provide a linear model (matrix factorization based) and a non-linear model (deep neural network based) for community modeling in recsys.

Extensive experiments on seven real-world datasets have been conducted in comparison with 14 state-of-art recommendation algorithms. Statistically significant improvements verify the effectiveness of the proposed implicit community based models. They also show superior performances in cold-start scenarios, which contributes to the application of real-life recommender systems.

Keywords: Recommender systems · Implicit community · Gibbs sampling

1 Introduction

Recommender systems try to analyze the user behaviors and provide information or items of interest to relevant users. Existing works have made efforts to model

A. Bouguettaya et al. (Eds.): WISE 2017, Part II, LNCS 10570, pp. 387–402, 2017.
DOI: 10.1007/978-3-319-68786-5_31

the user behaviors from the individual perspective and achieved great successes [10,17]. While human behaviors are the co-product of individual characteristics and community influences, it is necessary to model the impact of communities from the rating behaviors of users. In the context of recommender systems, the rating behaviors of users inside a same community are assumed to be more consistent, which is the basis of our proposed implicit communities in recommender systems.

In most recommender systems, there are no explicitly-labeled communities nor user-user connections. The available records only contain user-item interactions such as ratings or consumptions. Meanwhile, the user profiles such as ID, occupations and addresses can not directly reflect the preferences. Furthermore, the user profiles may not be publicly available considering the information security issues. For some recommender systems, the labeled communities usually come from real-life relationships. The users who are connected in real life may have diverse preferences (considering that they may simply come from same companies or universities but like different items). As the implicit communities should reflect user rating preferences, these explicitly labeled communities may not satisfy this requirement. In this case, we propose to learn the implicit communities directly from the rating records of users.

Therefore, we propose a bayesian probabilistic model that depicts the implicit communities. The Latent Factor Model is used as the cornerstone: user and items are modeled as latent factor vectors. In our model, the latent factor vectors of users inside a same community share a same distribution so that the behaviors of users are naturally influenced by the community. We use the co-clustering approach as another cornerstone to construct the implicit communities for users and items, where each community has a corresponding distribution of latent factor vectors. The community effect of items is considered because items from the same categories tend to have similar features (like romance movies and love novels may be both attractive to some youngsters), which has been convinced in works related to co-clustering methods [3,18].

The contributions of this paper are summarized as follows:

- To the best of our knowledge, We are about the first to model implicit communities for recommender systems, in terms of user rating behaviors.
- A unified probabilistic bayesian graphical model and corresponding parameter inference algorithms are proposed to discover the structures of implicit communities.
- The performances of proposed methods are compared with 14 state-of-the-art approaches on seven real-world datasets, and statistical significant improvements are observed. The proposed model is also verified to be effective in cold start cases, which contributes to the application of recommender systems in real-life scenarios.

The remainder of the paper is organized as follows. The next section introduces some related works and Sect. 3 introduces some preliminaries about the paper. Section 4 gives a detailed introduction about the modeling of implicit community from rating behaviors and further shows the inference algorithm for the parameters in the model. We conduct extensive experiments with the real-world datasets and the results are presented in Sect. 5. Finally, we conclude the paper in Sect. 6.

2 Related Work

2.1 Collaborative Filtering and Matrix Factorization Approaches

Collaborative Filtering (CF) is a typical approach for recommendation [17]. The motivation comes from the assumption that people often get the best recommendations from someone with tastes similar to themselves. Among various CF methods, Matrix Factorization (MF) is the most popular and effective one, which assumes that users and items are represented as vectors in a latent factor space. Some MF based approaches, including SVD++ [10], NMF (Non-Negative Matrix Factorization) [21], MMMF (Max-Marginal Matrix Factorization) [14], BMF (Biased Matrix Factorization) [10] and PMF (Probabilistic Matrix Factorization) [15] have achieved superior accuracy and scalability in recommendation due to the dimension reduction nature.

2.2 Localized Matrix Factorization Approaches

Co-clustering is widely used in image processing and bio-informatics. [18] is the first study related to applying co-clustering in recommender systems, which assumes that the matrix is generated from a bayesian probabilistic model. An additive co-clustering model is proposed in [2] where the matrix is assumed to be a summation of a series of matrices and each of them is co-clustered into blocks.

CF methods that utilize localized blocks include [22] and [3]. In [22], the matrix is first decomposed into several blocks along the diagonal and matrix factorization is performed in each sub-matrix later. [3] co-clusters the matrix into blocks but predict the missing entries in a different way from matrix factorization.

Another state-of-art approach is LLORMA [11]. It first randomly selects a number of user/item pair from the rating matrix, termed anchor points, and then chooses neighbors for the anchor points based the user and item similarities between the neighbors and the anchor point. Then the matrix factorization is performed on each submatrix and they are combined as a approximate the original rating matrix. In LLORMA, the anchor points decide the structure of localized matrices, but the random selection is ad-hoc and the matrix factorization is performed after the submatrices are determined. Our model discovers the structure of implicit communities and conducts the matrix factorization simultaneously. Meanwhile, the implicit communities are determined with a probabilistic model, which does not require the selection of anchor points artificially.

2.3 Community Detection

There has been some works concerning with the community detection or community discovery problems. In [19], these works are categorized into several aspects: Latent Space Models [4], Spectral Clustering [5] and Modularity Maximization [6]. The Latent Factor Models assume that the connections between users are determined by the latent factors. The community is discovered by clustering the users represented by latent factors. The spectral clustering method treats the social connections as an adjacency matrix of the social network and aims to minimize the number of connections between communities. This problem is a variant of minimum cut problem and can be solved with a spectral clustering method. The modularity maximization problem defines a metric for evaluating the quality of a partitioning of a network as modularity. A typical method for modularity maximization problem is greedy algorithm that starts with representing each node as a community and merges two communities with maximum incremental modularity.

3 Preliminaries

3.1 Latent Factor Model

Latent Factor Model (LFM) is widely adopted to describe the rating behavior of users: for user i and item j, the rating given by user i to item j is assumed to be a product of two latent factor vectors:

$$R_{ij} = u_i^T v_j + \tau \tag{1}$$

where u_i refers to the user factor vector which reflects the preferences of users; v_j refers to item factor vectors which reflects the qualities of items. τ is the global bias of ratings. The dimension of these vectors is a predefined constant where each dimension corresponds to a latent factor. The product of u_i and v_j therefore reflects the preference of user i on item j.

PMF (Probabilistic Matrix Factorization) has provided a good probabilistic interpretation for LFM as Fig. 1(a): it assumes the factor vectors are drawn from Gaussian distributions, where (μ_u, Σ_u) and (μ_v, Σ_v) represent the parameters of Gaussian distributions for the user and item latent factors respectively:

$$u_i \sim N(\mu_u, \Sigma_u), \forall i \in U,\ v_j \sim N(\mu_v, \Sigma_v), \forall j \in I \tag{2}$$

Among different variations of Latent Factor Model, PMF has an advantage of both good probabilistic interpretation and high accuracy, which is popularly used in related researches.

3.2 List of Symbols

We list the variables from our models in Table 1.

Table 1. Variables for the model proposed in this paper

Variable	Meaning
U	Set of users in the system, $\lvert U\rvert = N$
V	Set of items in the system, $\lvert V\rvert = M$
R_{ij}	Rating that user i gives to item j
c_i	The cluster that user i is in
g_j	The group that item j is in
u_i	User factor vector of user i
v_j	Item factor vector of item j
τ	The global bias
$(\mu_{c_i}, \Sigma_{c_i})$	Gaussian distribution parameters of u_i in cluster c_i
$(\mu_{g_j}, \Sigma_{g_j})$	Gaussian distribution parameters of v_j in group g_j
$\theta_\alpha : (\lambda_\alpha, \mu_\alpha, W_\alpha)$	parameters from prior of $(\mu_{c_i}, \Sigma_{c_i})$
$\theta_\beta : (\lambda_\beta, \mu_\beta, W_\beta)$	Parameters from prior of $(\mu_{g_j}, \Sigma_{g_j})$
θ	Parameter of multinomial distribution of c_i
γ	Parameter of multinomial distribution of g_j
σ	Precision of Gaussian distribution of R

3.3 Implicit Communities

Although each individual user/item is unique and different from each other, there exist different groups of users who have similar preferences on items. Notice that there are no explicit communities labeled or set up in the recommender systems, we call these naturally-formed similar-preferences user groups as **implicit communities**. Similarly, the implicit communities can be found in items based on users' preferences on them.

Notice that the users inside same implicit communities share similar preferences, they are not necessarily acquainted with each other in real life. The communities are observed from user favors on items (represented by ratings) in the system. This is another reason we call the communities as implicit ones.

4 Modeling Implicit Communities with Rating Behaviors

In this section, we present the generative model of implicit communities generated with rating behaviors. Moreover, the Gibbs sampling algorithms are presented for the parameter inference.

4.1 Implicit Community Coordinated Recommendation Model (ICR Model)

Given the effectiveness of latent factor models, we leverage the model to find the structure of implicit communities. The latent vectors of users/items inside the same communities share the same distribution. As the communities are not explicitly labeled, we design a probabilistic model capturing the rating behaviors and learning the implicit structure, which is called **Implicit Community coordinated Recommendation Model (ICR Model)** and shown in Fig. 1(b) in comparison with **PMF** (where the differences are highlighted).

ICR differs from PMF in two aspects: first, the latent factor vectors are drawn from different Gaussian distributions from the communities respectively. The assignment of users/items to the communities is learned simultaneously with the latent factor vectors; second, we put priors over the Gaussian distributions and the parameters of these Gaussian distributions are learnt iteratively.

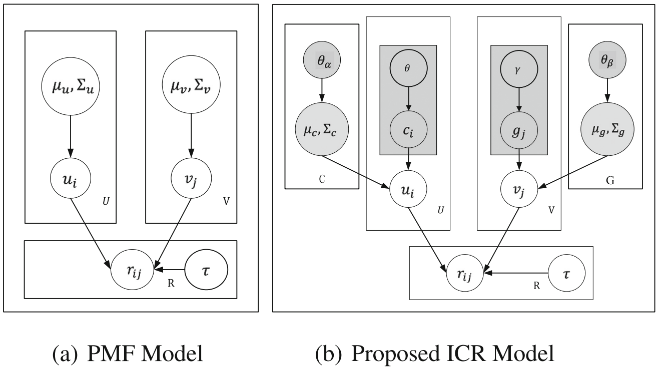


Fig. 1. Comparison between PMF and ICR Model

The reason for choosing Gaussian-Wishart distribution is that it is conjugate to the Gaussian distribution, which allows for a convenient inference of the parameters. Meanwhile multinomial distribution and Gaussian distribution are chosen to represent the community membership and rating distribution respectively.

Here is the generation process of the latent factor vector for user i:

- *Choose the hyperparameters:* $\lambda_\alpha, \mu_\alpha, W_\alpha$;
- *For each user, generate the cluster belonging to with the multinomial distribution:* $c_i \sim Multi(\theta)$;
- *For each cluster, generate the corresponding Gaussian parameters:* $(\mu_c, \Sigma_c) \sim GW(\lambda_\alpha, \mu_\alpha, W_\alpha)$;
- *For each user, generate the user factor vector with the Gaussian distribution from the cluster* c_i: $u_i \sim N(\mu_{c_i}, \Sigma_{c_i})$;

Unlike the users, items may not possess the subjective rating behaviors by choosing to be rated by whom. However, we believe that community effect still exist objectively, this can be explained that some categories of items may be rated similarly: the book "Game of Thrones" and the movie "Lord of the Ring" may be close to each other for some group of users like both of them; or some other items will be put into a same community because they are liked/disliked by a group of users. Therefore we model the implicit communities of items in a similar way to users by the generation process:

– *Choose the hyperparameters:* $\lambda_\beta, \mu_\beta, W_\beta$;
– *For each item, generate the group it belongs to with the multinomial distribution:* $g_j \sim Multi(\gamma)$;
– *For each group, generate the corresponding Gaussian parameters:* $(\mu_g, \Sigma_g) \sim GW(\lambda_\beta, \mu_\beta, W_\beta)$;
– *For each item, generate the item factor vector with the Gaussian distribution from the group* g_j: $v_j \sim N(\mu_{g_j}, \Sigma_{g_j})$;

4.2 Implicit Community Coordinated Recommendation Algorithm

Given the parameter inference procedures introduced, we present the algorithm for model learning in Algorithm 1.

Algorithm 1. Implicit Community coordinated Recommendation Algorithm (ICR):

Input: The rating matrix R, the number of user clusters $|c|$ and the number of item groups $|g|$
Output: Implicit clusters of users c, implicit groups of items g and predictions of unknown ratings.

```
Initialize the latent factor vectors u and v;
while Not convergent and iter ≤ MaxIter do
    for each user i and item j do
        Infer the user cluster c_i and the item group g_j;
    end for
    for each cluster c and group g do
        Infer the parameters for the prior μ_c, Σ_c, μ_g, Σ_g;
    end for
    for each user i and item j do
        Infer user and item factor vectors: u_i, v_j;
    end for;
end while
Predict the unknown ratings by drawing from the normal distribution respectively;
```

The ICR algorithm first initializes the user and item factor vectors, then assigns the users and items to different implicit communities according to the cluster and group assignment inference procedure (as Eqs. (8), (9)). When the implicit communities are found, the parameters of Gaussian distributions from each cluster and group are inferred (as Eq. (11)). With the inferred parameters, the factor vectors can be updated (as Eqs. (3), (5)). Then the ratings can be predicted with the factor vectors inferred. The process keeps going in iterations and outputs the final implicit communities of users and items with predictions of unknown ratings.

Inference for u_i, v_j. Based on our proposed model, the conditional distribution of u_i on the rest of parameters is still a Gaussian.

$$
\begin{aligned}
p(u_i|rest) =& N(u_i|\mu^*, \Sigma^{*-1}) \\
&\sim \prod_{j=1}^{M} [N((R_{ij} - \tau)|u_i^T v_j, \sigma^{-1})]^{I_{ij}} p(u_i|\mu_{c_i}, \Sigma_{c_i}^{-1})
\end{aligned}
\tag{3}
$$

where

$$
\begin{aligned}
\Sigma^* &= \Sigma_{c_i} + \sigma \sum_j [v_j v_j^T]^{I_{ij}} \\
\mu^* &= [\Sigma^*]^{-1} (\sigma \sum_j [v_j (R_{ij} - \tau)]^{I_{ij}} + \Sigma_{c_i} \cdot \mu_{c_i})
\end{aligned}
\tag{4}
$$

I_{ij} is an indicator variable that indicates if user i has rated item j (1 if rated and 0 otherwise). c_i represents the cluster user i is in. The conditional distribution of v_j on the rest of parameters is a Gaussian as well:

$$
\begin{aligned}
p(v_j|rest) =& N(v_j|\mu^*, \Sigma^{*-1}) \\
&\sim \prod_{j=1}^{M} [N((R_{ij} - \tau)|u_i^T v_j, \sigma^{-1})]^{I_{ij}} p(v_j|\mu_{g_j}, \Sigma_{g_j}^{-1})
\end{aligned}
\tag{5}
$$

where

$$
\begin{aligned}
\Sigma^* &= \Sigma_{g_j} + \sigma \sum_i [u_i u_i^T]^{I_{ij}} \\
\mu^* &= [\Sigma^*]^{-1} (\sigma \sum_i [u_i (R_{ij} - \tau)]^{I_{ij}} + \Sigma_{g_j} \cdot \mu_{g_j})
\end{aligned}
\tag{6}
$$

g_j represents the group item j is in.

Inference for c_i and g_j. Now we evaluate how well a user/item is fit for the cluster/group assigned to and introduce how we assign users and items into clusters and groups. The likelihood of u_i and v_j conditioning on that user i is assigned to the cluster $c_i = \hat{c}$ (where $\hat{c}$ refers to a specific cluster) and that item j is assigned to the group $g_j = \hat{g}$ (where $\hat{g}$ refers to a specific group) are:

$$
\begin{aligned}
p(u_i|c_i = \hat{c}, \mu_{c_i}, \Sigma_{c_i}) &\sim N(u_i|\mu_{\hat{c}}, \Sigma_{\hat{c}}^{-1}) \\
p(v_j|g_j = \hat{g}, \mu_{g_j}, \Sigma_{g_j}) &\sim N(v_j|\mu_{\hat{g}}, \Sigma_{\hat{g}}^{-1})
\end{aligned}
\tag{7}
$$

Since $p(c_i = \hat{c}|u_i, \mu_{\hat{c}}, \Sigma_{\hat{c}}) \sim p(u_i|c_i = \hat{c}, \mu_{c_i}, \Sigma_{c_i})$, so the probability of user i assigned to $\hat{c}$ is:

$$
p(c_i = \hat{c}) = \frac{N(u_i|\mu_{\hat{c}}, \Sigma_{\hat{c}}^{-1})}{\sum_c N(u_i|\mu_c, \Sigma_c^{-1})}
\tag{8}
$$

Similarly, the group assignment can be inferred as this:

$$
p(g_j = \hat{g}) = \frac{N(v_j|\mu_{\hat{g}}, \Sigma_{\hat{g}}^{-1})}{\sum_g N(v_j|\mu_g, \Sigma_g^{-1})}
\tag{9}
$$

Inference for $\boldsymbol{\mu_c}, \boldsymbol{\Sigma_c}$ and $\boldsymbol{\mu_g}, \boldsymbol{\Sigma_g}$. The priors for the Gaussian distributions of cluster parameters are drawn from the GW (Gaussian-Wishart) distribution. The posterior distribution of the parameters is a GW distribution as well:

$$\begin{aligned} p(\mu_c, \Sigma_c|rest) &= GW(\mu_c, \Sigma_c|\lambda^*, \nu^*, \mu^*, W^*) \\ &= N(\mu_c|\mu^*, (\lambda^* \Sigma_c)^{-1})\, W(\Sigma_c|W^*, \nu^*) \end{aligned} \tag{10}$$

where

$$\begin{aligned} &\lambda^* = \lambda_\alpha + N_c, \nu^* = \nu_0 + N_c \\ &\mu^* = \lambda^{*-1}(\lambda_\alpha \mu_\alpha + \sum_{c_i=c} u_i) \\ &W^{*-1} = W_\alpha^{-1} + \sum_{c_i=c}(u_i - \bar{u}_c)(u_i - \bar{u}_c)^T + \\ &\frac{\lambda_\alpha N_c}{\lambda_\alpha + N_c}(\mu_\alpha - \bar{u}_c)(\mu_\alpha - \bar{u}_c)^T \\ &\bar{u}_c = \frac{\sum_{c_i=c} u_i}{N_c} \end{aligned} \tag{11}$$

N_c denotes the number of users assigned to cluster c. Similarly, the parameters $\mu_g, \Sigma_g, \forall g$ can be drawn from the GW distributions with updated parameters.

5 Experiments

In this section, we conduct the experiments on real-world datasets to evaluate the performances of our algorithm. More specifically, we first introduce the datasets and evaluation metrics adopted in our experiments, then the results on rating-only datasets are presented to evaluate the performances of ICR. Then the performances of ICR are compared with other social-aware recommender systems on social-included datasets. Moreover the results on cold-start users are presented.

5.1 Datasets

In our experiments, we use the datasets from Movielens-100K, Movielens-1M, Movielens-10M, Film Trust, Ciao, Epinions and Douban. The first three datasets are datasets from the website Movielens and widely adopted in the evaluation of recommender systems, however they do not contain social relationships. The latter four datasets come from the review websites that allow users to post their reviews for the items online and contain in-site social relationships. The first three datasets contain in-site trust relationships between users, which are unilateral, i.e. the user may trust the other one but the trusted user does not necessarily trust back. Douban contains friendship between users, which is bilateral, i.e. users are friends with each other.

The details of the datasets are presented in Table 2. The density of ratings in Moivelens-100K, Movielens-1M and Movielens-10M are 6.30%, 4.19% and

Table 2. Details of the datasets

Datasets	#users	#items	#ratings	#trusts
M.L.-100K	943	1,682	100,000	\
M.L.-1M	6,040	3,952	1,000,209	\
M.L.-10M	69,878	10,677	10,000,054	\
Film trust	1,508	2,071	35,497	1,632
Ciao	7,375	106,796	282,269	111,781
Epinions	40,163	139,738	664,824	487,183
Douban	129,490	58,541	16,830,839	1,692,952

1.34% respectively. The densities of ratings and trusts in FilmTrust are 1.14% and 0.052%; while the densities of ratings and trusts in Ciao are 0.037% and 0.21%; the Epinion dataset has a rating density of 0.051% and a trust density of 0.029%. For Douban, the rating and social densities are 0.22% and 0.01%. For each dataset, we split them into 5 folds and use 4 folds as training set while the remaining fold is used as testing set. The average performances are presented in the paper.

5.2 Benchmarks and Evaluation Metrics

We choose 14 benchmark algorithms for comparison with our algorithms ICR, including: *Rating-Only Recommendation Algorithms*:

- **PMF** [15]**:** PMF is a probabilistic model that generates user and item factor vectors from Gaussian distributions.
- **BMF** [10]**:** BMF (Biased Matrix Factorization) includes user and item biases into the matrix factorization model. The addition of bias improves the accuracy of prediction.
- **BPMF** [16]**:** BPMF (Bayesian Probabilistic Matrix Factorization) places priors on the user and item latent factor vectors into the Probabilistic Matrix Factorization.
- **NMF** [8]**:** NMF (Non-negative Matrix Factorization) is another matrix factorization scheme that requires the latent factor vectors to be non-negative.
- **SVD++** [10]**:** SVD++ is a model that merges latent factor model and neighbourhood effect together. Furthermore, it can be extended to incorporate both implicit and explicit feedbacks from users.

Localized Matrix Factorization and Clustering based Approach (also rating only):

- **BCC (Bayesian Co-Clustering)** [18]**:** The BCC algorithm co-clusters the matrix into several blocks so that the entries inside the same cluster have a low variance.
- **LMF** [22]**:** LMF (Localized Matrix Factorization) first decomposes the matrix into several blocks and then conduct matrix factorization in each sub-matrix.

- **LLORMA(Local Low-Rank Matrix Approximation)** [11]**:** It is a state-of-the-art matrix factorization model based on the local low rank assumption.
- **UCMF(User Clustering Matrix Factorization)** [1]**:** It is a graphical model that clusters users into K groups for recommendation.
- **ICMF(Item Clustering Matrix Factorization)** [1]**:** It is a graphical model that clusters users into K groups for recommendation, as opposite to the UCMF recommender.

Social-Aware Recommendation Algorithms:

- **SoRec** [12]**:** Sorec co-factorizes the rating matrix and social matrix simultaneously and both matrices share the same user factor vectors.
- **SoReg** [13]**:** The model adds social regularization into the matrix factorization framework based on the social homophily effect.
- **SocialMF** [9]**:** The model employs matrix factorization techniques as the basis and incorporates the mechanism of trust propagation into the model.
- **TrustMF** [20]**:** TrustMF assigns each user a trustor-specific vector and a trustee-specific vector. The model can choose to incorporate either vector or both vectors in the matrix factorization framework.

Proposed Implicit Community Models:

- ***ICR:*** This is the Implicit Community coordinated Recommendation Algorithm proposed in this paper.

Since there are no public available toolkits for LMF found, we select their best performances reported from their papers in following result tables. The results of BCC are generated with the tool from [18]. The results of remaining baselines are conducted with LibRec [7]. We select social-aware recommendation algorithms because social connections are explicitly labeled connections and we want to compare the performances of ICR with algorithms that incorporate explicit user relationships.

We use RMSE for the rating prediction evaluation:

$$RMSE = \sqrt{\frac{1}{T}\sum_{i,j}(R_{ij} - \hat{R}_{ij})^2} \tag{12}$$

R_{ij} denotes the actual rating user i gives to item j, $\hat{R}_{ij}$ denotes the predicted R_{ij}, and T is the size of testing set.

Parameter setting: In our algorithm, we set $\sigma = 2$ for all the datasets. The hyperparameters of the Gaussians are set as: $(\lambda_\alpha, \mu_\alpha, W_\alpha) = (0, \mathbf{0}, \frac{3}{2}I)$, where I is the indentity matrix and the settings are the same for θ_β and θ_ρ. The numbers of clusters and groups for different datasets are set as (2,2), (5,5), (10,10), (4,5), (6,8), (6,10), (10,8) (in order of Movielens-100K, Movielens-1M, Movielens-10M, FilmTrust, Ciao, Epinions and Douban). The numbers are selected by a grid search in experiments.

The parameters for benchmark algorithms are chosen from the reported ones in the references and further carefully tuned.[1]

5.3 Comparative Analyses on Proposed Approaches

Performances of ICR. The performances of our algorithm and other non-social aware recommendation algorithms on Movielens datasets are summarized in Table 3. As shown in the table, our method outperforms the other approaches and achieves comparative performance with LLORMA. However, LLORMA requires a selection of anchor points beforehand and the localized matrices are divided based on the selected anchor points. Similarly in LMF model, the whole matrix is decomposed into several blocks where each block is a sub-matrix consisting of a portion of rows (users) and columns (items), then the standard matrix factorization is conducted on each sub-matrix. Both LLORMA and LMF artificially segment the matrix into localized matrices first and conduct matrix factorization separately. This separation of localization and matrix factorization does not directly incorporate user preferences into the localization and therefore can not capture the structure of implicit communities.

Table 3. RMSE on movielens datasets (ICR Imp. is the improvement of ICR on others)

Methods	ML-10M	ICR Imp.	ML-1M	ICR Imp.	ML-100K	ICR Imp.
UCMF	0.978	19.63%	1.035	18.84%	1.042	14.59%
ICMF	0.944	16.74%	0.979	14.20%	1.025	13.17%
PMF	0.819	4.03%	0.871	3.44%	0.960	7.29%
BPMF	0.816	3.68%	0.865	3.10%	0.954	6.71%
BMF	0.806	2.48%	0.879	4.44%	0.916	2.84%
NMF	0.824	4.61%	0.881	4.65%	0.914	2.63%
LMF	\	\	0.866	3.11%	0.910	2.20%
SVD++	0.803	2.46%	0.867	3.11%	0.912	2.41%
BCC	0.985	20.20%	1.051	20.08%	1.062	16.20%
LLORMA	0.789	-	0.840	-	0.894	-
ICR	**0.786***	-	**0.840***	-	**0.890***	-

ICR incorporates the clustering of users and items and latent factor model together into a generative model and learn the structure of implicit communities and latent factor vectors simultaneously. This enables a coordination of user preference into the community discovery and judging from the experimental results, this procedure improves the rating prediction accuracy (Table 4).

[1] In all of the following result tables, $*$ represents the improvements of ICR are statically significant with $p < 0.05$.

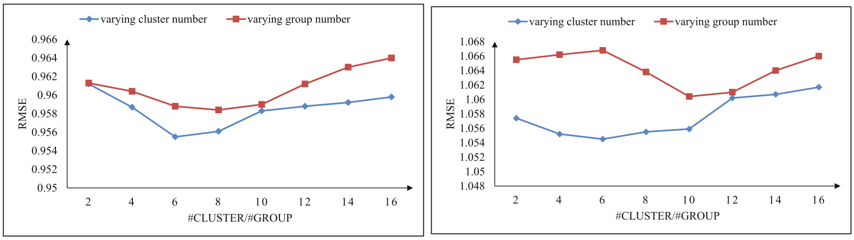
We further conduct experiments on social-included datasets and compare the prediction accuracy of ICR with other state-of-art approaches, including social-aware recommendation algorithms. Since the social connections are explicitly labeled relationships, the social-aware recommendation algorithms usually achieve superior performances than those non-social algorithms. However, we find that ICR performs best among all the comparative approaches, which illustrates the contribution of modeling implicit communities on the rating prediction task. As the social connections provide good complementary information to alleviate the shortage of ratings, the social-aware approaches can utilize social information to model user preferences. On the other hand, ICR does not rely on social information and only utilize ratings to learn the structure of implicit communities from rating records. The implicit communities directly reflects the user preferences while the social connections are not strongly correlated to the rating similarities of users. Therefore the superior performance of ICR shows the effectiveness of implicit communities.

Table 4. RMSE comparison with social-aware algorithms

Methods	PMF	NMF	BMF	SVD^{++}	SoRec	SoReg	SocialMF	TrustMF	BCC	ICR
FilmTrust	0.968	0.974	0.856	0.802	0.831	0.875	0.844	0.819	0.831	**0.788***
ICR Imp.	18.60%	19.10%	7.94%	1.75%	5.17%	9.94%	6.64%	3.79%	5.17%	-
Ciao	1.076	1.264	1.006	0.983	1.014	1.078	0.978	1.012	1.014	**0.964***
ICR Imp.	10.41%	23.73%	4.17%	1.93%	4.93%	10.58%	1.43%	4.74%	4.93%	-
Epinions	1.197	1.302	1.107	1.067	1.142	1.095	1.082	1.095	1.186	**1.053***
ICR Imp.	12.03%	19.12%	4.88%	1.31%	7.79%	3.84%	2.68%	3.84%	11.21%	-
Douban	0.720	0.723	0.722	0.712	0.753	0.700	0.774	0.724	0.768	**0.694***
ICR Imp.	3.61%	4.01%	3.88%	2.53%	7.84%	0.86%	10.34%	4.14%	9.64%	-

Impacts of Clusters and Groups. The experiment is conducted to reveal the impact of cluster and group number on the performances. We fix the cluster number to 1 and check the results with various group numbers (for group number impact it is symmetric). The results are similar on all datasets. Due to page limit, the results on dataset Ciao and Epinion are presented. As illustrated in Fig. 2, the performance on Ciao gets better when the number of clusters rises from 2 to 6, however, the performance does not keep getting better with the number of clusters growing (such as 8 to 12 clusters). When the number of clusters is properly limited, the growing number of clusters can assign users to proper clusters. Therefore the user factor vectors can be generated with more sufficient information from these users inside the cluster. When the number of clusters is large, the data assigned to train each cluster is 'diluted', and the user factor vectors can not be generated accurately, which leads to the decrease of the prediction precision. Therefore, we need to find the suitable number of clusters to maximize the prediction precision, and the number is 6 to 8 for Ciao.

The impact of group numbers is quite similar to the impact of cluster numbers. The same phenomenon appears in the experiments on Epinions and the results are depicted in Fig. 2.



(a) Impact on Ciao (b) Impact on Epinions

Fig. 2. Impact of cluster and group numbers

5.4 Performances in Cold-Start Scenarios

In this section, we present the results in cold-start scenarios. Users who rate fewer than five items are referred as cold-start users and the evaluation is conducted only on these users here. Similar testing settings are used in [20]. The number of ratings for each user in Movielens datasets is at least 20, therefore no cold-start users exist in these datasets.

Table 5. RMSE comparison on cold-start users

Methods	PMF	NMF	BMF	SVD++	SoRec	SoReg	SocialMF	TrustMF	ICR
Film trust	1.009	0.904	1.421	0.898	0.914	0.973	0.934	0.913	**0.884***
ICR Imp.	12.39%	2.21%	37.79%	1.56%	3.28%	9.15%	5.35%	3.18%	-
Ciao	1.191	1.046	1.327	1.020	1.033	1.278	1.017	1.031	**1.007***
ICR Imp.	15.45%	3.73%	24.11%	1.27%	2.52%	21.21%	0.98%	2.33%	-
Epinions	1.432	1.197	1.412	1.166	1.180	1.437	1.152	1.176	**1.126***
ICR Imp.	21.37%	5.93%	20.25%	3.43%	4.58%	21.64%	2.26%	4.25%	-
Douban	0.827	0.828	0.826	0.827	0.833	0.815	0.839	0.840	**0.809***
ICR Imp.	2.18%	2.29%	2.06%	2.18%	2.88%	0.74%	3.58%	3.69%	-

The cold start problem is a coherent trouble in recommender systems since users usually rate a considerably small number of items. Meanwhile, new users and items are added to the system all the time, which provides little training data for the recommender system. The cold-start problem can be alleviated from the implicit community effect in ICR while social-aware recommendation algorithms utilize social information to enrich the user profiles of 'cold' users.

As shown in Table 5, ICR still achieves a superior performance over the benchmarks, even when comparing with other social-aware recommender systems. In ICR, the cold-start users with few ratings are assigned to corresponding clusters based on the similarity of their preferences to users inside. Therefore the preferences of these users are enriched by the profiles of other users in the corresponding clusters. The results indicate that the implicit communities provides more accurate modeling for the preferences of cold users. As the social connections provide an enrichment of user profiles in a different perspective, we conjecture that joint modeling of implicit communities and social connections in recommender systems can further improve the prediction accuracy. We will leave it as one of the future work directions.

6 Conclusions

In this paper, we concern with the problem of how to model implicit communities in recommender systems and further utilize the communities to improve the performances of recommender systems. We design a bayesian generation probabilistic model that detects the implicit communities from the rating records. Moreover, we design a Gibbs sampling algorithm for parameter inference. Extensive experiments have been conducted on 7 real world datasets and the results in comparison with 13 state-of-the-art approaches show statistically significant improvements. To the best of our knowledge, we are the first to model the implicit communities in recommender systems based on the rating behaviors of users. In the future, we aim to jointly model implicit communities and social connections for recommendation.

Acknowledgement. This work is supported in part by China Grant U1636215, 61572492, and the Hong Kong Scholars Program.

References

1. Barbieri, N., Manco, G., Ritacco, E.: Probabilistic approaches to recommendations. Synth. Lect. Data Min. Knowl. Discov. **5**(2), 1–197 (2014)
2. Beutel, A., Ahmed, A., Smola, A.J.: ACCAMS: additive co-clustering to approximate matrices succinctly. In: Proceedings of the 24th International Conference on World Wide Web, pp. 119–129. International World Wide Web Conferences Steering Committee (2015)
3. Beutel, A., Murray, K., Faloutsos, C., Smola, A.J.: CoBaFi: Collaborative bayesian filtering. In: Proceedings of the 23rd International Conference on World Wide Web, WWW 2014, pp. 97–108. ACM, New York (2014)
4. Borg, I., Groenen, P.J.F.: Modern multidimensional scaling: theory and applications. J. Educ. Measur. **40**(3), 277–280 (2003)
5. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: KDD 2001, pp. 269–274. ACM, New York (2001)
6. Good, B.H., De Montjoye, Y., Clauset, A.: The performance of modularity maximization in practical contexts. Phys. Rev. E **81**(4), 46106 (2009)
7. Guo, G., Zhang, J., Yorke-Smith, N.: TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 25–30 January 2015, Austin, Texas, USA, pp. 123–129 (2015)
8. Hoyer, P.O.: Non-negative matrix factorization with sparseness constraints. J. Mach. Learn. Res. **5**, 1457–1469 (2004)

9. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, 26–30 September 2010, pp. 135–142 (2010)
10. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434. ACM (2008)
11. Lee, J., Kim, S., Lebanon, G., Singer, Y., Bengio, S.: LLORMA: local low-rank matrix approximation. J. Mach. Learn. Res. **17**(15), 1–24 (2016)
12. Ma, H., Yang, H., Lyu, M.R., King, I.: SoRec: social recommendation using probabilistic matrix factorization. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 931–940. ACM (2008)
13. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 287–296. ACM (2011)
14. Rennie, J.D., Srebro, N.: Fast maximum margin matrix factorization for collaborative prediction. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 713–719. ACM (2005)
15. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: NIPS, vol. 1, pp. 1–2 (2007)
16. Salakhutdinov, R., Mnih, A.: Bayesian probabilistic matrix factorization using markov chain monte carlo. In: Proceedings of the 25th International Conference on Machine Learning, pp. 880–887. ACM (2008)
17. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, pp. 285–295. ACM (2001)
18. Shan, H., Banerjee, A.: Bayesian co-clustering. In: Eighth IEEE International Conference on Data Mining, ICDM 2008, pp. 530–539. IEEE (2008)
19. Tang, L., Liu, H.: Community detection and mining in social media. Synth. Lect. Data Min. Knowl. Discov. **2**(1), 1–137 (2010)
20. Yang, B., Lei, Y., Liu, D., Liu, J.: Social collaborative filtering by trust. In: IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013 (2013)
21. Zhang, S., Wang, W., Ford, J., Makedon, F.: Learning from incomplete ratings using non-negative matrix factorization. In: SDM, vol. 6, pp. 548–552. SIAM (2006)
22. Zhang, Y., Zhang, M., Liu, Y., Ma, S.: Improve collaborative filtering through bordered block diagonal form matrices. In: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2013, pp. 313–322. ACM, New York (2013)