Nonparametric Bayesian Probabilistic Latent Factor Model for Group Recommender Systems

Chapter · November 2016

DOI: 10.1007/978-3-319-48740-3_5

Xiongcai Cai
UNSW Sydney

Nonparametric Bayesian Probabilistic Latent Factor Model for Group Recommender Systems

Nipa Chowdhury✉, Xiongcai Cai, and Cheng Luo

School of Computer Science and Engineering, The University of New South Wales, Sydney NSW 2052, Australia
{nipac,xcai,luoc}@cse.unsw.edu.au

Abstract. The explosion of the online web encourages online users to participate in group activities. Group recommender systems are essential for recommending items to a group of users based on their common preferences. However, existing group recommender systems do not exploit user interaction within a group and merely work on groups with fixed sizes of users and same levels of similarity among group members, which significantly limits its usage in real world scenarios. In this paper, we propose a novel nonparametric Bayesian probabilistic latent factor model to learn the collective users' tastes and preferences for group recommendation by exploiting user interaction within a group, which is able to well handle a variety of group sizes and similarity levels. We evaluate the developed model on three publicly available benchmark datasets. The experimental results demonstrate that our method outperforms all baseline methods for group recommendation.

Keywords: Group recommender systems, collaborative filtering, Bayesian probabilistic matrix factorisation, Dirichlet prior

1 Introduction

Recommender systems (RS) [3,4,9] have been recognised as a personal assistant to online users to find relevant information/products to satisfy their personal preferences. Matrix factorisation (MF) is a popular method to build personalised recommender systems, which shows great success in the Netflix prize competition [9]. The major purpose of MF in RS is to obtain some forms of lower-rank approximation to the original feedback matrix for understanding the interaction between user preferences and item attractiveness in forms of latent factors.

Although the RS are primarily designed to satisfy individual preferences, there are scenarios where a group of users are interested in participating in a single activity, for example, in entertainment purpose, watching a movie, playing an online video game, listening music and visit a tourist place. The group recommender systems (GRS) generate a single recommendation list for a group of users, aiming to maximise all users satisfaction [13]. The rapid explore of online data allows users to find people who have a common interest and participate in group activities, which draws the attention of researchers on GRS. However, the group recommendation scenario is more challenging than the individual one as users have different preferences and different levels of interaction within a group [1]. MF algorithms that are designed for personalised RS [9, 15], failed to learn latent factors to model collective user tastes and preferences, thus are not suitable for group recommendation [6].

Existing research on GRS can be divided into two categories [1,2,4,7,8,13,18]. The first one is based on profile aggregation where a virtual user is created by averaging all user profiles in a group [12, 18]. The recommendation list for the group is generated by considering only the virtual user. The second category is based on rank aggregation where a recommendation list for the group is generated by merging all user individual recommendation lists [1, 2, 13]. Personalised RS methods are used to generate the individual recommendation. However, the contribution of both profile aggregation and rank aggregation-based group recommendation systems is limited on the aggregation technique. The performances of these methods also depend on the size of the group and the level of similarity among group members. Furthermore, the main drawback in the existing methods is that they always ignore the interaction among the group members in generating recommendations, whereas the interaction among members may make a great effect in group decisions.

In this paper, a novel method named nonparametric Bayesian probabilistic latent factor model (NBPLFM) is proposed for group recommender systems where the task is to generate recommendations for a group of users. The proposed method extends Bayesian probabilistic matrix factorisation (BPMF) [15] to consider group of users tastes and preferences, by applying a Dirichlet process mixture model to the prior of the user and item latent factor. Rather than relying on the aggregation strategy like existing GRS, the innovative NBPLFM models group user preferences by exploiting their interaction within the group. As a result, this proposed model explicitly takes into account collective users preferences in learning and generating recommendations. Further, the integration of nonparametric prior allows NBPLFM to model group with variable group size and similarity, which reflects the real world group recommendation scenario. To verify the performance, we apply the proposed method on three public datasets. The experimental results confirm the efficiency of the proposed method compared to other baseline methods for group recommendation.

The rest of the paper is arranged as follows: in Sect. 2, we summarise related work. Section 3 presents the proposed matrix factorisation method. Experimental results are presented in Sect. 4. Finally, we draw conclusions in Sect. 5.

2 Related Work

The approaches that are adopted to build GRS can be divided into two categories, i.e., profile aggregation and rank aggregation [1, 2, 4, 7, 8, 13, 18]. Rank aggregation methods come with better flexibility than profile aggregation methods and the results are explainable. Different merging strategies [4, 11] from social theory such as additive utilitarian, multiplicative utilitarian, average, least misery, average without misery and most happiness are employed to aggregate individual recommendation lists.

The authors in [2] study different types of aggregation strategies and report that the performance of the GRS does not depend on the aggregation strategies, but depends on the similarity among users within a group and also on the group sizes. Polylens [13] is developed as a group recommender to recommend movies to Movielens[1] users. It uses the nearest neighbourhood algorithm to derive an individual recommendation list. The group recommendation list is generated by merging the individual recommendation lists according to least misery approach. Further, the research in [1] introduces the notion of consensus function, that aims at maximising item relevance and minimising disagreements between group members. It uses the average or least misery approach to compute group relevance. To compute group disagreement between members, it proposes two alternative ways: average of pairwise disagreements or score variance. Recently, some model-based approaches are proposed for GRS [16, 17, 19]. For example, a consensus model named COM is proposed for generating group recommendations by assuming that item selection in a group depends not only on a user's personal choice of the content factors but also on group topics [19] . The research in [17] believes that social influence of friends also contributes to one's item selection decision. Based on this assumption, it proposes a probabilistic generative model by exploiting the preferences of group members and their pairwise influence on each other to reach the final decision. The authors in [16] use the k-means algorithm on the user latent factors to detect user groups. They create group latent factors by averaging the user latent factor and used with item latent factors to generate group recommendations. Because contextual and social information is sensitive and expensive to collect, we avoid to use such information and thus do not compare our method with other methods [17, 19] that use contextual or social information in addition to user feedback.

Although, the above methods improve the group recommendation performances but these methods have some limitations for widely commercial applications. For instance the profile aggregation methods [12,18] and rank aggregation methods [1, 2, 13] in group recommender systems rely on the aggregation strategies. They do not account user interaction within a group and thus failed to model collective user preferences. User preference within a group is different from individual user preference and the RS should capture the users combined preference to generate group recommendations. Moreover, most of the existing research perform well when the groups are small in size (consists of 2,3,5,8 or 10 users) and have high similarity [1, 2, 13]. However, social influence [17] and topic influence [19] of group members do not hold when groups are large. Therefore it is important to learn the collective preferences of users, especially for a large group which is still a challenging open question in group recommendation.

To overcome these limitations, this study proposes a novel method named nonparametric Bayesian probabilistic latent factor model for group recommendation by learning user and item latent factor to reflect tastes and preferences of the group. We extend BPMF by imposing Dirichlet process mixture model as a prior of user and item latent factor. The users latent factors are learned to

[1] https://movielens.org/

maximise the overall group satisfaction by exploiting user interaction within a group. Modelling users interaction in learning enables our model to generate recommendations for groups with any sizes and similarity levels. Thus, NBPLFM is able to improve performance significantly compared to the other approaches to group recommendation.

3 Nonparametric Bayesian Probabilistic Latent Factor Model

In this section, we firstly present a key component of our algorithm named Bayesian probabilistic matrix factorisation. Then we extend it with the nonparametric prior distribution of user and item latent factors to learn the factors that are best suitable for group recommendations.

3.1 Bayesian Probabilistic Matrix Factorisation (BPMF)

Assuming there are N users and M items in the data, let matrix $R \in \Re^{N\times M}$ be a user preference matrix. In MF, $R \in \Re^{N\times M}$ can be approximated by two low rank matrices $U \in \Re^{D\times N}$ and $V \in \Re^{D\times M}$ as $R \approx U^{'}V$ by minimising the sum of squared errors, where $D \ll min(N, M)$ is the dimensionality of user latent factors and item latent factors [9].

BPMF [15] employs a probabilistic linear model with Gaussian noise to learn the user latent factor and item latent factor. It also places prior and hyperprior over the model parameters and hyperparameters, respectively. It assumes that the user parameters (U) and item parameters (V) follow the multivariate Gaussian distribution and the user hyperparameters (θ_u) and the item hyperparameters (θ_v) follow the Gaussian-Wishart distribution. Figure 1(a) shows the graphical model of BPMF. The conditional distribution over the observed ratings and the prior distributions over the user and item parameters and hyperparameters are given by:

$$\begin{aligned}
\mathcal{P}(R \mid U, V, \sigma) &= \prod_i \prod_j [\mathcal{N}(R_{ij} \mid U_i^{'} V_j, \sigma)],\\
\mathcal{P}(U \mid \theta_u) &= \prod_i \mathcal{N}(U_i \mid \mu_u, \Lambda_u),\\
\mathcal{P}(V \mid \theta_v) &= \prod_j \mathcal{N}(V_j \mid \mu_v, \Lambda_v),\\
\mathcal{P}(\theta_u \mid \theta_0) &= \mathcal{N}(\mu_u \mid \mu_0, (\beta_0 \Lambda_u)^{-1}) \mathcal{W}(\Lambda_u \mid \mathcal{W}_0, v_0),\\
\mathcal{P}(\theta_v \mid \theta_0) &= \mathcal{N}(\mu_v \mid \mu_0, (\beta_0 \Lambda_v)^{-1}) \mathcal{W}(\Lambda_v \mid \mathcal{W}_0, v_0),
\end{aligned}$$

where $\theta_0 = \{\mu_0, \mathcal{W}_0, v_0, \beta_0\}$ and $\mathcal{W}(\Lambda|\mathcal{W}_0, v_0)$ is the Wishart distribution with v_0 degrees of freedom and scale matrix W_0. The posterior predictive distribution of a rating R^*_{ij} is obtained by marginalizing over model parameters and hyperparameters.

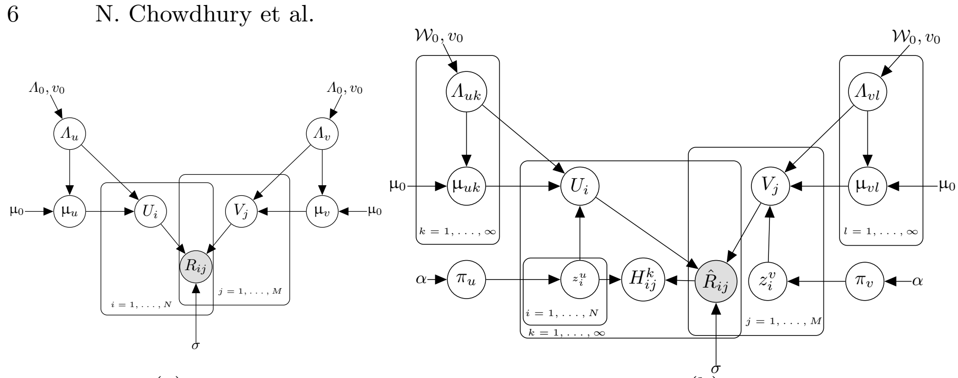
$$\mathcal{P}(R^*_{ij} \mid R, \theta_0) = \int\int \mathcal{P}(R^*_{ij} \mid U_i, V_j) \mathcal{P}(U, V \mid R, \theta_u, \theta_v) \mathcal{P}(\theta_u, \theta_v \mid \theta_0).$$

The computation of this predictive distribution is analytically intractable; thus, Markov chain Monte Carlo (MCMC) is used to approximate inference. Specifically, Gibbs sampler is used to calculate the approximation. BPMF shows promising results in personalised recommendation without heavily relying on parameter tuning. It assumes that the user parameter and item parameter are drawn from a single common multivariate normal distribution. However, in GRS, we assume that the users within a group are more similar than the population in general. Items that shares same genre are more similar than the rest of the items. Thus, using a common distribution over all user parameters and hyper-parameters limits the performance of BPMF on group recommendation. In this regard, we propose NBPLFM for group recommendation.

3.2 Nonparametric Bayesian Probabilistic Latent Factor Model

NBPLFM is based on the observation that the users and items within a group should be more similar than other group or the rest of the population. It draws the user and item latent factors from their group specific distributions which enables the resultant model to capture group choice more effectively and coherently. In reality, the number of the groups and the size of each group should not be fixed in group recommendation because a recommender system does not know before hand the number of users who will form a group to watch a movie, music or to visit a tourist spot. Similarly, the system does not know in advance how many groups will be built to consume a product together. So the group recommendation model should allow the variability of group size as well as the flexibility for creating the number of groups. Existing research such as [1, 2, 13] assumes a fixed number of groups and fixed number of users within each group in group formulation, which does not comply with the real world scenarios of group recommendation. To integrate the variability of group size and the adaptability of group number into our model, we exploit a nonparametric distribution and apply it over the user and movie latent factors. Specifically, we use a Dirichlet process mixture model that does not require the prior knowledge of the size of the groups. The proposed model allows an infinite number of mixture components; thus, the model is able to introduce new mixture components when new users are added [5]. The research in [14] also uses a nonparametric prior for the user latent class and item latent class, which uses side information about users and items to achieve an improved personalised recommendation performance. However, the method in [14] does not model the interaction among users within each group, thus is unable to learn collective users tastes and preferences in group recommendation scenario.

The probabilistic graphical model of our method is shown in Fig. 1(b). We assume that the users belong to k distinct cluster with means μ_u and items are distributed to l different cluster with means μ_v. The value of k could be infinite and automatically decided by our method. The prior probability of user cluster π_u and item cluster π_v are distributed according to a Dirichlet distribution $Dir(.)$. Cluster means μ_u and μ_v are generated from Gaussian distribution and cluster variances Λ_u and Λ_v are generated from Wishart distribution $\mathcal{W}(.)$. The user cluster label z^u and item cluster label z^v are sampled from their corresponding multinomial distribution $Multi(.)$. Please note that we transform the observed ratings R_{ij} into Gaussian distributions to reduce the rating variance across different user and item groups. The conditional distribution over the transformed



(a) (b)

Fig. 1: Graphical model for (a) Bayesian probabilistic matrix factorisation and (b) Nonparametric Bayesian probabilistic latent factor model.

observed ratings $\hat{R_{ij}}$ and the prior distribution over the user and item parameters and their group specific hyperparameters are given below:

$$\mathcal{P}(\hat{R} \mid U, V, \sigma) = \prod_i \prod_j \mathcal{N}(\hat{R}_{ij} \mid U_i^{'} V_j, \sigma), \tag{1}$$

$$\mathcal{P}(H \mid \hat{R}_{ij}, z_i^u) = \Delta(\hat{R}_{ij}; z_i^u), \tag{2}$$

$$\mathcal{P}(U \mid \theta_{uk}, z^u) = \prod_i \mathcal{N}(U_i \mid \mu_{z_i^u}, \Lambda_{z_i^u}), \tag{3}$$

$$\mathcal{P}(V \mid \theta_{vl}, z^v) = \prod_j \mathcal{N}(V_j \mid \mu_{z_j^v}, \Lambda_{z_j^v}), \tag{4}$$

$$\mathcal{P}(\theta_{uk} \mid \theta_0) = \mathcal{N}(\mu_{uk} \mid \mu_0, \Lambda_{uk}/\beta_0)\mathcal{W}(\Lambda_{uk} \mid \mathcal{W}_0, v_0), \tag{5}$$

$$\mathcal{P}(\theta_{vl} \mid \theta_0) = \mathcal{N}(\mu_{vl} \mid \mu_0, \Lambda_{vl}/\beta_0)\mathcal{W}(\Lambda_{vl} \mid \mathcal{W}_0, v_0), \tag{6}$$

$$\mathcal{P}(z^u \mid \pi_u) = Dir(\alpha/k), \tag{7}$$

$$\mathcal{P}(z^v \mid \pi_v) = Dir(\alpha/l), \tag{8}$$

where $\theta_{uk} = \{\mu_{uk}, \Lambda_{uk}\}$ and $\{\theta_{vl} = \mu_{vl}, \Lambda_{vl}\}$. In a group, an item may be rated by many users. Average and least misery are commonly used aggregation techniques in group recommendation literature to learn the group's preferences for an item [7]. However, the least misery approach ignores the majority of users' information by concentrating on the user that has the smallest preference [4]. To account all users preferences for an item we use an average of user rating to learn the group's preference on that item. Thus, Δ refers as average function. We use Chinese restaurant process (CRP) to sample user membership and item membership from the DP mixture model with K components where the limit of the component K goes to infinity. According to CRP, the Nth user will be assigned to an existing group with probability $\frac{N_k}{N-1+\alpha}$ and it will be assigned to a new group with probability $\frac{\alpha}{N-1+\alpha}$, where α is a positive scaling parameter and N_k is the number of users already exists in group k [5]. The generative process for clusters is shown as follows:

1) generates cluster distributions for each user and for each item:
   $\pi_u \sim Dir(\alpha/k)$,
   $\pi_v \sim Dir(\alpha/l)$;
2) generates mean and variance for each user cluster:

$\mu_{uk} \sim \mathcal{N}(\mu_0, \Lambda_{uk}/\beta_0)$,
$\Lambda_{uk} \sim \mathcal{W}(v_0, \mathcal{W}_0)$;

3) generates mean and variance for each item cluster:
$\mu_{vl} \sim \mathcal{N}(\mu_0, \Lambda_{vl}/\beta_0)$,
$\Lambda_{vl} \sim \mathcal{W}(v_0, \mathcal{W}_0)$;

where $k = 1, \ldots, K$ and $l = 1, \ldots, L$. After determining the cluster parameter, NBPLFM follows the following generative process to sample the cluster label parameter, the user and item latent factor parameter and the rating:

1) for each user $i = 1, \ldots, N$ and for each item $j = 1, \ldots, M$, sample the user cluster label and item cluster label:
$z_i^u \sim Multi(\pi_u)$,
$z_j^v \sim Multi(\pi_v)$;
2) for groups of users $k = 1, \ldots, K$, samples their vectors of parameters:
$U_i^k \sim \mathcal{N}(U_i \mid \mu_{z_i^u}, \Lambda_{z_i^u})$;
3) for each item $j = 1, \ldots, M$, sample a vector of parameters:
$V_j \sim \mathcal{N}(V_j \mid \mu_{z_j^v}, \Lambda_{z_j^v})$;
4) for users in group k , calculate
$H_{ij}^k \sim \Delta(\hat{R}_{ij}^k; z^u)$;
5) for each user i and movie j, sample a rating:
$\hat{R}_{ij} \sim \mathcal{N}(\hat{R}_{ij} \mid U_i^{'} V_j, \sigma)$.

We formulate the user cycle group wise and item cycle item wise since our goal is to recommend items to various user groups. The posterior predictive distribution of a rating R_{ij}^* is obtained by marginalising over model parameters and group specific hyperparameters:

$$\mathcal{P}(R_{ij}^* \mid R, \theta_0) = \int\int \mathcal{P}(R_{ij}^* \mid U_i, V_j)\mathcal{P}(U \mid R, H, V, \theta_{uk})\mathcal{P}(V \mid R, U, \theta_{vl})\mathcal{P}(\theta_{uk} \mid \theta_0)$$
$$\mathcal{P}(\theta_{vl} \mid \theta_0)\mathcal{P}(z^u \mid \pi_u)\mathcal{P}(z^v \mid \pi_v).$$

Since the computation of above equation is expensive, following [15] we use Gibbs sampler to calculate an approximation. The Gibbs sampler cycles through the latent variables U_i, V_j, θ_{uk}, θ_{vl}, z_i^u and z_j^v are conditioned on current values of other variables. The conditional distribution of U_i is

$$\mathcal{P}(U_i \mid \hat{R}, H, V, \theta_{uk}, z^u) \propto \mathcal{P}(U_i \mid \theta_{uk}, z^u)\mathcal{P}(H_{ij}^k \mid \hat{R}_{ij}; z^u) \prod_{ij} \mathcal{N}(H_{ij}^k \mid U_i^{'} V_j, \sigma)$$
$$\mathcal{P}(U_i \mid \hat{R}, H, V, \theta_{uk}, z^u) = \mathcal{N}(U_i \mid \mu_{uk}^*, \Lambda_{uk}^*), \tag{9}$$

where ${\Lambda_{uk}^*}^{-1} = \Lambda_{uk}^{-1} + \frac{1}{\sigma^2}\sum_j V_j V_j^{'}$, and $\mu_{uk}^* = \Lambda_{uk}^*(\Lambda_{uk}^{-1}\mu_{uk} + \frac{1}{\sigma^2}\sum_j [V_j H_{ij}])$. The conditional distribution over the group specific user hyperparameters conditioned on user factor matrix are given by the Gaussian-Wishart distribution:

$$\mathcal{P}(\mu_{uk}, \Lambda_{uk} \mid U, \theta_0) = \mathcal{N}(\mu_{uk} \mid \mu_a, \Lambda_{uk}/\beta_a)\mathcal{W}^{-1}(\Lambda_{uk} \mid \mathcal{W}_a, v_a), \tag{10}$$

where $\mu_a = \frac{\beta_0}{\beta_0 + N_{uk}}\mu_0 + \frac{N_{uk}}{\beta_0 + N_{uk}}\overline{U}$, $\mathcal{W}_a = \Lambda_0 + S + \frac{\beta_0 N_{uk}}{\beta_0 + N_{uk}}(\overline{U} - \mu_0)(\overline{U} - \mu_0)^{'}$, $\beta_a = \beta_0 + N_{uk}$, $v_a = v_0 + N_{uk}$ and $S = \sum_i^{N_{uk}}(U_i - \overline{U})(U_i - \overline{U})^{'}$. The group membership of user and items are sampled according to the following equations:

Algorithm 1 Nonparametric Bayesian probabilistic latent factor model

Initialise model parameter U^1, V^1
for t=1:T **do**
1 Sample group membership of users and items according to:
$z_i^u \sim Multi(\pi_u)$,
$z_j^v \sim Multi(\pi_v)$.
2 Sample the hyperparameters of user and item group according to (10):
$\theta_{uk}^t \sim \mathcal{P}(\theta_{uk} \mid U, \theta_0)$,
$\theta_{vl}^t \sim \mathcal{P}(\theta_{vl} \mid V, \theta_0)$.
3a For users in a group k, sample U_i^k according to (9):
$U_i^{t+1} \sim \mathcal{P}(U_i \mid \theta_{uk}, z^u, V, H, \hat{R})$.
3b For each item in a group, sample V_j according to (12):
$V_j^{t+1} \sim \mathcal{P}(V_j \mid \theta_{vl}, z^v, \hat{R}, U)$.
end for

$$\begin{aligned}\mathcal{P}(z_i^u = k \mid z_{-i}^u, U_i, \theta_u, H) &= b.\frac{N_{uk}^{-i}}{N_u - 1 + \alpha}\mathcal{N}(U_i \mid \theta_{uk}),\\ \mathcal{P}(z_i^u \neq z_j^u forallj \neq i \mid z_{-i}^u, U_i, \theta_u) &= b.\frac{\alpha}{N_u - 1 + \alpha}\\ \int \mathcal{N}(U_i \mid \theta_u^*)\mathcal{N}(\mu_{nu}^* \mid \mu_0, \Lambda_v/\beta_0)&\mathcal{W}^{-1}(\Lambda_{nv}^* \mid \mathcal{W}_0, v_0)d(\theta_u^*)),\end{aligned} \tag{11}$$

where N_u is the number of users, N_{uk} the number of users assigned to group k i.e $N_{uk} = \sum_i [z_i^u = k]$ and $\theta_u^* = \{\mu_{nu}^*, \Lambda_{nv}^*\}$. The conditional distribution of V_j is

$$\begin{aligned}\mathcal{P}(V_j \mid \hat{R}, U, \theta_{vl}, z^v) &\propto \mathcal{P}(V_j \mid \theta_{vl}, z^v)\prod_{ij}\mathcal{N}(\hat{R}_{ij} \mid U_i^{'}V_j, \sigma),\\ \mathcal{P}(V_j \mid \hat{R}, U, \theta_{vl}, z^v) &= \mathcal{N}(V_j \mid \mu_{vl}^*, \Lambda_{vl}^*),\end{aligned} \tag{12}$$

where ${\Lambda_{vl}^*}^{-1} = \Lambda_{vl}^{-1} + \frac{1}{\sigma^2}\sum_i U_i U_i^{'}$ and $\mu_{vl}^* = \Lambda_{vl}^*(\Lambda_{vl}^{-1}\mu_{vl} + \frac{1}{\sigma^2}\sum_i [U_i \hat{R}_{ij}])$. The computation of $\mathcal{P}(\theta_{vl} \mid V, \theta_0)$ follows the same equation as (10). The computation of z^v also follows (11), except it does not include H in its computation. The sampling algorithm of NBPLFM is given in Algorithm 1.

4 Experiments

4.1 Dataset

NBPLFM is evaluated on three publicly available benchmark datasets for group recommendation including 1) Movielens 100K,[2] denoted as ML, to recommend movies to a group of user; 2) Amazon video game dataset,[3] denoted as VG, to recommend video games to a group of players and 3) Amazon digital music dataset,[3] denoted as DM, to recommend music to a group of listeners. ML dataset consists of 100,000 ratings from 943 users on 1682 movies. VG dataset consists of 463,668 ratings from 228,570 users/reviewers on 21,025 video games.

[2] http://www.grouplens.org/node/73
[3] McAuley, J., Pandey, R., Leskovec, J.: Inferring networks of substitutable and complementary products. In: KDD '15. pp. 785–794 (2015)

For 99,128 ratings in this dataset, we observe 'unknown' value in the userid field. As 'unknown' does not mean any particular user, we remove those entries from the dataset. Again, for some video games we find multiple reviews from the same reviewer. In this case, we omit the preceding reviews and only count the latest review. Following the widely accepted procedure [10], the users who rated fewer than 5 movies and the items that have been rated by less than 3 users are removed. After the preprocessing, the dataset consists of total 60,913 ratings from 5644 users on 4568 video games. DM dataset consists of total 836,015 ratings from 478,243 users on 266,416 music. Following [10], we further remove users that review less than 10 songs and songs that are rated by fewer than 5 users resulting in 45,045 ratings from 4209 users on 2926 music. Users preferences on items are expressed as ratings on all datasets and scaled on integers 1-5.

In group recommendation literature, groups are formed considering different sizes and similarity levels. The majority of the research [1, 2, 8, 11] consider groups containing a small number of users such as 2, 3, 5, 8 or 10 users and also account pre-defined similarity among the users. In reality, we cannot pre-define how many users will form a group. It may limit the method performance on groups with specific size and similarity by creating a restriction on group member size and similarity level. In our research, groups are formed without the restriction of the group size and pre-define similarity and only various numbers of groups are considered. More specifically, we use k-means algorithm to create G different groups. For example, when G=20, there will be 20 different groups of users. In our experiment, G contains various ranges of values. The smaller value of G corresponds to less groups, where each group contains a large but indefinite number of users. Whereas the larger value of G refers to more groups, where each group includes smaller but indefinite number of users. For ML dataset we set $G \in \{4,5,10,15,20,25,30,40,50,60,70,80,90,100\}$, and for VG and DM datasets we set $G \in \{10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160\}$. As ML dataset contains fewer user than VG and DM dataset respectively, smaller ranges of values for G is used to carry experiment on this dataset. Each group has a variable number of users with varying levels of similarity.

Each dataset is divided into a training set and a test set for each group. Each training set consists of 80% ratings of the members of a group and the remaining 20% ratings are used for test. Following [2], we only include items in the test set that are not present in the training set of any members for a group. We generate 10 versions of the dataset by randomly sampling items and the average result is reported. We also use 20% data from training as the validation set to tune the model parameters.

4.2 Evaluation Metric

Following the standard evaluation metric used in [2, 19], we use normalised discounted cumulative gain (NDCG) as IR performance measure for the testing and evaluation of our method. We compare the recommendation list generated for a group with the recommendation list generated from ground truth ratings for the group. As the datasets do not contain explicit group preference information, we adopt an aggregation method to compute the group's preference on items. We are not aware of any other research in group recommendation literature that generates recommendations for group that contains more than 10 users. However, the author in [8] reports that additive utilitarian is the winning strategy for the larger group when comparing various aggregation method vs. group size. Following [4, 7, 8, 11], we also use the additive utilitarian strategy to compute group's preference on items. According to this strategy, each member's preference to test items are added and the recommendation list for the group is generated by sorting the summed score. In summary, NDCG score for a group g can be computed as

$$NDCG_g@n = \frac{DCG_g@n}{IDCG_g@n},$$

$$DCG_g@n = \sum_{j=1}^{n} \frac{2^{R_j^g - 1}}{log_2(j+1)},$$

where j is the item that appears at the j th position in the top-n recommendation list for the group g and group score can be computed by $R_j^g = \sum_{u \in g} R_{uj}$.

4.3 Compared Methods

We compare the performance of NBPLFM with a number of group recommendation approaches as follows:

1) UCF-AVG User-based CF is the widely used approach which analyses the similarities among users to predict the recommendation preference on unseen user-item pairs. In group recommendation settings, it first calculates user's preferences on unseen items by following user-based CF method and averages of these predictions to generate group recommendation score for that item.

2) UCF-MIS The method uses user-based CF method to generate individual predictions and least-misery approach to calculate group recommendation score.

3) UCF-RD This model calculates the recommendation score on an unseen item based on both relevance and disagreement between group members [1]. After computing individual prediction by user-based CF method, the average aggregation strategy is applied in the model to measure the relevance and score variance to measure the disagreement.

4) LFM This model uses a latent factor model for group recommendation [16].

5) BPMF It is the state-of-the-art method of recommendation [15].

6) BPMFDP BPMF with Dirichlet process mixture prior [14]. In our implementation of this method, we do not consider side information as such information is sensitive and expensive to collect as mentioned in Section 2.

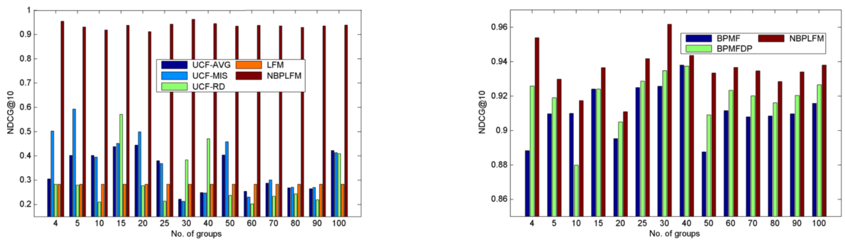
4.4 Parameter Setting

For UCF-AVG, UCF-MIS and UCF-RD, we compute similarity only to the users who have at least 5 items in common. Following the best experimental results in [1], we also weighted relevance score by 0.2 and disagreement by 0.8 to generate group recommendation score. For each algorithm, we tune parameters separately on a validation set. Following the best performance of BPMF, BPMFDP and NBPLFM on validation set, we set latent factor dimension of user and item matrix D=10, observation noise σ=0.9, μ_0=0, v=0 and $\mathcal{W}_0$=0 for each methods on all datasets. On ML and DM datasets, we use α=0.0013 for both BPMFDP and NBPLFM. The value α=0.013 is used for both NBPLFM and BPMFDP on VG dataset. Since our method is based on Gibbs sampler, the first few samples from the Markov chain are discarded (burn-in), as they may not represent the desired distribution. From the validation burn-in samples are 20, 30 and 50 for ML, DM and VG, respectively. The average over the next 50, 70 and 150 samples from the posterior predictive distribution are used to generate results on ML, DM and VG datasets respectively.

4.5 Results

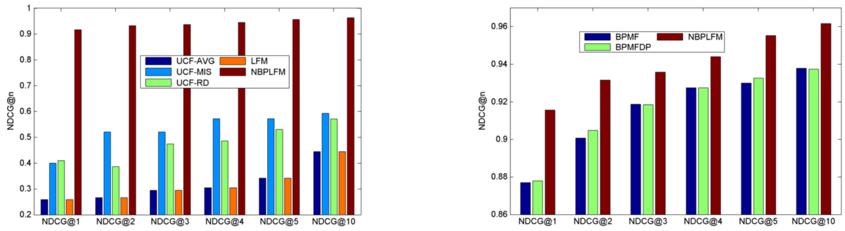
To verify the proposed method, we apply it with compared methods on ML dataset and the NDCG@10 performances results on this dataset are presented in Fig. 2. We also examine the above methods performances by varying top-n recommendation size. The best performance of these methods for NDCG@n=1,2,3, 4,5,10 on ML dataset is reported in Fig. 3.

From this bar diagram as shown in Fig. 2, it is clear indicated that NBPLFM achieves significant improvement overall performances than that of other methods for all group numbers. From Fig. 2(a), it reaches 33∼74% performance improvement over both UCF-AVG and UCF-MIS. It also scores 36∼72% performance improvement when the comparison is made with both UCF-RD and LFM. As shown in Fig. 2(b), our method also improves nearly 0.74∼7% performance improvement over BPMF. In comparison with BPMFDP, our method achieves maximum improvement amount to 3.74% for G=10. It also achieves 0.59∼2.45% improvement for all other group numbers. It is worthy noted that both BPMF and BPMFDP achieves the best performance for G=40, while the proposed NBPLFM exhibits the best performance for G=30. UCF-AVG, UCF-MIS, LFM and UCF-RD achieves the best performance for group number 20, 5, 4 and 15, respectively. As shown in Fig. 3(a) NBPLFM achieves more than 50∼65% improvement over UCF-AVG and LFM for different top-n positions. Over UCF-MIS and UCF-RD, it gains 36∼51% and 39∼54% improvement re-



(a) (b)

Fig. 2: NDCG@10 comparison on NBPLFM with (a) UCF-AVG,UCF-MIS,UCF-RD and LFM and (b) with BPMF and BPMFDP on ML dataset.

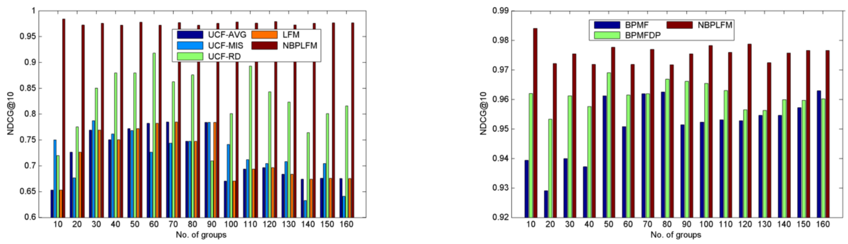


(a) (b)

Fig. 3: Performance comparison on NBPLFM with (a) UCF-AVG,UCF-MIS,UCF-RD and LFM and (b) with BPMF and BPMFDP for various top-n positions on ML dataset.

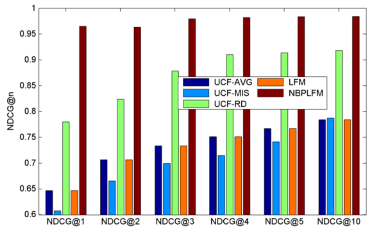
spectively. From Fig. 3(b), NBPLFM shows better performance among all other algorithms. In numbers, our method gains 1.64∼3.85% improvement over BPMF and 1.64∼3.76% improvement over BPMFDP for various top-n recommendation position. The reasons for NBPLFM outperforming all compared methods will be discussed at the end of this section.

The NDCG@10 performance of the methods on the VG dataset is shown in Fig. 4 and the best performance of the methods w.r.t varying top-n positions are shown in Fig. 5. From the result in Fig. 4(a), it is found that NBPLFM achieves 18∼33% performance improvement over both UCF-AVG and UCF-MIS. Over UCF-RD, it achieves highest 26% improvement for G=90 and the lowest improvement is 8% for G=110. It also gains more than 65% improvement for all group number over LFM. From Fig. 4(b), NBPLFM gains 0.92∼4.5% performance improvement over BPMF. Our method also improves the performance of BPMFDP by 0.5∼2.2%. It is obvious from Fig. 5 that our method has better performance than other baseline methods for all top-n positions. UCF-AVG, LFM, UCF-MIS, UCF-RD and NBPLFM model achieves the best performance when the value of G is 90, 90, 30, 60 and 10, respectively. Over UCF-AVG and UCF-MIS, NBPLFM achieves 19∼35% improvement for all NDCG@n computations in Fig. 5(a). In comparison with LFM, our method achieves more than 65% improvement for all top-n positions. It also gains 7∼10% improvement over UCF-RD. BPMF and BPMFDP achieve the best performance for G=80 and G=50, respectively. Comparing to BPMF and BPMFDP in Fig. 5(b), our method achieves more than 5% performance improvement on NDCG@1 computations. For all other NDCG@n

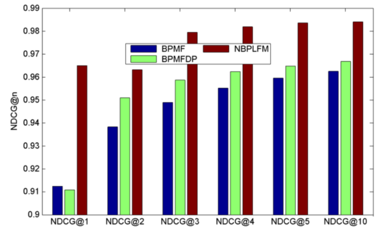


(a) (b)

Fig. 4: NDCG@10 comparison on NBPLFM with (a) UCF-AVG,UCF-MIS,UCF-RD and LFM and (b) with BPMF and BPMFDP on VG dataset.



(a)



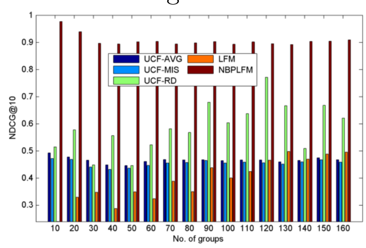
(b)

Fig. 5: Performance comparison on NBPLFM with (a) UCF-AVG,UCF-MIS,UCF-RD and LFM and (b) with BPMF and BPMFDP for various top-n positions on VG dataset.

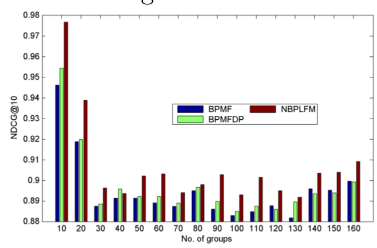
computations, it gains 2.15∼3% and 1.22∼2.18% improvement over BPMF and BPMFDP, respectively.

The results on DM dataset are plotted in Fig. 6 and 7. Over UCF-AVG, UCF-MIS and LFM, our method achieves more than 40% improvement in Fig. 6(a) for all group number. In comparison with UCF-RD, NBPLFM achieves 22∼46% improvement. Comparing to BPMF, it achieves more than 0.3∼3% improvement in Fig. 6(b). Over BPMFDP, our method achieves nearly 2% improvement for G=10 and G=20. Although for G=40, our method performs slightly worse than BPMFDP and for all other cases it gains almost 0.25∼1.4% performance improvement. UCF-AVG, UCF-MIS, BPMF, BPMFDP and NBPLFM model achieves the best performance when the value of G is 10 on this dataset. LFM and UCF-RD show the best performance for G=120 and G=160, respectively. NBPLFM achieves more than 39% improvement in top-n recommendation over UCF-AVG, UCF-MIS and LFM in Fig. 7(a). Over UCF-RD, it increases the performance more than 20% for all top-n positions. From Fig. 7(b), it gains 0.5∼4.72% improvement over BPMFDP and 1∼5.68% improvement over BPMF for all top-n positions.

Overall, these results indicate the effectiveness of NBPLFM is better as compared to the other state-of-the-art approaches. The reasons for the experimental results can be explained as follows. UCF-AVG and UCF-MIS use average and least misery strategy to generate group recommendations. They do not model group information in learning and the results are obvious. UCF-RD exploits both relevance and disagreement between group members to generate recommenda-

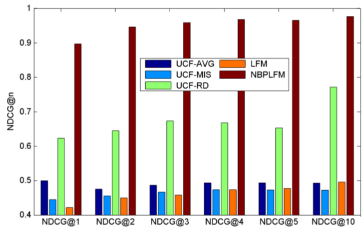


(a)

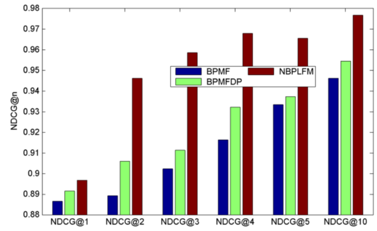


(b)

Fig. 6: NDCG@10 comparison on NBPLFM with (a) UCF-AVG,UCF-MIS,UCF-RD and LFM and (b) with BPMF and BPMFDP on DM dataset.



(a)



(b)

Fig. 7: Performance comparison on NBPLFM with (a) UCF-AVG,UCF-MIS,UCF-RD and LFM and (b) with BPMF and BPMFDP for various top-n positions on DM dataset.

tions. As this information is not modelled into learning, the method does not score well. The results indicate their sensitivity with groups where user to user similarity varies from high to low and also with groups where the members size are not fixed and can vary from one group to another group. The LFM model uses a model-based approach to generate individual recommendation and average user's latent factors of a group to generate the group latent factor. However, this model does not consider the user interaction within a group and the average latent factor of a group may represent a virtual user who may not represent any members within the group. Thus, LFM is more sensitive for a group that consists of a large number of users. BPMF also does not exploit group information in learning, thus does not achieve satisfactory results in a group recommendation scenario. Although BPMFDP uses BPMF with Dirichlet prior, they do not exploit user interaction within a group in learning. To overcome these limitations the proposed method focuses on the user interaction within a group. It exploits such vital interactions by learning group-based latent factors that represent user preferences within each group and consider the collaborations within the group and from other groups. Thus, our model incorporates group preference in learning and also in generating recommendations. The use of infinite mixture model allows our model to learn group preference with variable group size and similarity. While the baseline methods performance are limited by group size and similarity, it is shown that the proposed model can generate recommendation regardless of any specific group size and similarity. Therefore the proposed model significantly improves the performance compared to the other state-of-the-art baseline methods for the task of top-n group recommendation.

5 Conclusion

In this paper, we propose a nonparametric Bayesian probabilistic latent factor model, namely NBPLFM, for group recommendations. Rather than relying on the aggregation method in generating recommendations, the proposed method learns user latent factors from their past interactions within the group. To fulfil the real-world group recommendation scenario, the proposed method has a freedom to formation of a group i.e., with variable group size and similarity. As a result, the performance of NBPLFM is not limited by the group size and similarity. Whereas, existing group recommender systems focuses on generating recommendations for groups that contain a smaller number of users which are fixed across all groups and also having the same level of similarity. NBPLFM follows nonparametric approach, thus comes with the flexibility of an infinite number of groups. Comparing experiment results on three public benchmark datasets verify the NBPLFM's efficiency to generate recommendations for groups with variable size and similarity. In this paper, we assume that a user belongs to only one group. In future work, we will extend our model to handle the situation where a user could belong to multiple groups.

References

1. Amer-Yahia, S., Roy, S.B., Chawla, A., Das, G.: Group recommendation: Semantics and efficiency. In: VLDB 2009 (2009)
2. Baltrunas, L., Makcinskas, T., Ricci, F.: Group recommendations with rank aggregation and collaborative filtering. In: RecSys '10. pp. 119–126 (2010)
3. Cai, X., Bain, M., Krzywicki, A., Wobcke, W., Kim, Y.S., Compton, P., Mahidadia, A.: Collaborative filtering for people to people recommendation in social networks. In: AI 2010: Advances in Artificial Intelligence. pp. 476–485. Springer Berlin Heidelberg (2010)
4. Cantador, I., Castells, P.: Recommender systems for the social web (2012)
5. Gelman, A., Robert, C., Chopin, N., Rousseau, J.: Bayesian data analysis (1995)
6. Hu, L., Cao, J., Xu, G., Cao, L., Gu, Z., Cao, W.: Deep modeling of group preferences for group-based recommendation. In: AAAI'14. pp. 1861–1867 (2014)
7. Jameson, A., Smyth, B.: Recommendation to groups. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web: Methods and Strategies of Web Personalization. pp. 596–627. Springer Berlin Heidelberg (2007)
8. Kompan, M., Bieliková, M.: Group recommendations: Survey and perspectives. Computing and Informatics 33(2), 446–476 (2014)
9. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer 42, 30–37 (2009)
10. Marlin, B.: Modeling user rating profiles for collaborative filtering. In: NIPS (2003)
11. Masthoff, J.: Group modeling: Selecting a sequence of television items to suit a group of viewers. User Modeling and User-Adapted Interaction 14(1), 37–85 (2004)
12. McCarthy, J., Anagnost, T.: Musicfx: An arbiter of group preferences for computer supported collaborative workouts. In: CSCW 1998. pp. 363–372 (1998)
13. O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J.: Polylens: A recommender system for groups of users. In: ECSCW'01. pp. 199–218 (2001)
14. Porteous, I., Asuncion, A.U., Welling, M.: Bayesian matrix factorization with side information and dirichlet process mixtures. In: AAAI (2010)
15. Salakhutdinov, R., Mnih, A.: Bayesian probabilistic matrix factorization using markov chain monte carlo. In: ICML '08. pp. 880–887. ACM (2008)
16. Shi, J., Wu, B., Lin, X.: A latent group model for group recommendation. In: IEEE MS 2015. pp. 233–238 (2015)
17. Ye, M., Liu, X., Lee, W.C.: Exploring social influence for recommendation: A generative model approach. In: SIGIR '12. pp. 671–680 (2012)
18. Yu, Z., Zhou, X., Hao, Y., Gu, J.: Tv program recommendation for multiple viewers based on user profilemergin. In: UMAP 2006 (2006)
19. Yuan, Q., Cong, G., Lin, C.: Com: a generative model for group recommendation. In: KDD '14. pp. 163–172 (2014)