Combining Latent Factor Model with Location Features for Event-based Group Recommendation

Wei Zhang[†], Jianyong Wang[‡], Wei Feng[†]
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[†]{zwei11,feng-w10}@mails.tsinghua.edu.cn, [‡]jianyong@tsinghua.edu.cn

ABSTRACT

Groups play an essential role in many social websites which promote users' interactions and accelerate the diffusion of information. Recommending groups that users are really interested to join is significant for both users and social media. While traditional group recommendation problem has been extensively studied, we focus on a new type of the problem, i.e., event-based group recommendation. Unlike the other forms of groups, users join this type of groups mainly for participating offline events organized by group members or inviting other users to attend events sponsored by them. These characteristics determine that previously proposed approaches for group recommendation cannot be adapted to the new problem easily as they ignore the geographical influence and other explicit features of groups and users.

In this paper, we propose a method called Pairwise Tag-enhAnced and featuRe-based Matrix factorIzation for Group recommendAtioN (PTARMIGAN), which considers location features, social features, and implicit patterns simultaneously in a unified model. More specifically, we exploit matrix factorization to model interactions between users and groups. Meanwhile, we incorporate their profile information into pairwise enhanced latent factors respectively. We also utilize the linear model to capture explicit features. Due to the reinforcement between explicit features and implicit patterns, our approach can provide better group recommendations. We conducted a comprehensive performance evaluation on real word data sets and the experimental results demonstrate the effectiveness of our method.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information Filtering, Retrieval Models*

Keywords

Event-based Group Recommendation, Location Feature, Latent Factor Model

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.
Copyright 2013 ACM 978-1-4503-2174-7/13/08 ...$15.00.

1. INTRODUCTION

Recently, social groups, which are normally self-organized communities, have emerged and become prevalent in many online social media, such as Flickr[1], Facebook[2], and Douban[3], etc. It is common that users prefer to join a group in which members share some common interest. Unlike direct relationship between users, the inherent characteristics of groups, e.g., focusing on some specific topics, determine that the relations are clearer. Hence, they can promote information diffusion and users' interactions in groups.

However, it is non-trivial for ordinary users to discover groups which they want to join for the following three reasons. First, the volume and variety of groups are overwhelming for users. Second, users often cannot express their preference accurately when they use vertical search engine to find groups. Moreover, users sometimes just want to be told which groups they prefer to join. In order to solve these problems, researchers proposed different methods for group recommendation according to users' personal needs [25, 27, 6]. Most of them are based on latent factor model with some variations, which can convert each user and group into a low dimensional representation and then calculate the degree of closeness between them.

In this paper, we focus on a new type of problem about recommending groups, i.e., personalized recommendation of event-based groups to users. Event-based social networks, which are mainly constituted by groups, have attracted more and more research interest recently [14, 15]. Different from groups mentioned before, event-based groups mainly include offline events, which means members will meet each other when they participate in events held by other members in the same group. Intuitively, friend relationship plays a more significant role in users' decisions to join this kind of groups, which needs to be addressed in this problem. Besides, the nature of real-world events indicates location information should be considered in the proposed methods as well. Yet, previous research methods in group recommendation do not exploit location features of users and groups.

Note that the authors of [15, 17] also used the term of 'group recommendation' to define their problems, however, their tasks are to recommend some specific items to a group of users, and their approaches concentrate on how to take all users' preferences in a group into account and automatically determine different decision influence weights of different users from the same group. While our work is about per-

[1] http://www.flickr.com/groups/

[2] http://www.facebook.com/about/groups/

[3] http://www.douban.com/group/explore

sonalized recommendation of event-based groups to a user, which is totally different from the problems studied in [15, 17]. Thus, their methods cannot be applied to our problem. For ease of representation, we define the notion of group recommendation in this paper as recommending groups to users to join but not recommending items to groups.

Based on the above analysis, we propose a new method for personalized event-based group recommendation. We consider both implicit and explicit factors that could influence users' decisions in our model. More specifically, we adopt latent factor model to capture interactions between users and groups like previous methods on group recommendation. Meanwhile, we incorporate their profile information into pairwise enhanced latent factors. We also employ the linear model to combine explicit features, including location features and social features of users and groups which are popularly used in location-based recommendation.

Our major contributions in this paper are summarized as follows:

(1) We propose a novel problem called personalized event-based group recommendation in a local city to help users find the groups they want to join. While ranking all groups in all cities for each user is low efficient, we exploit the localization property of users and groups, and convert the problem to recommending groups which are organized in the users' living cities, which is reasonable based on our analysis in Section 3.1. To our best knowledge, we are the first to study this problem.

(2) To take the advantage of the previous methods in traditional group recommendation problem and the location properties of event-based groups and users, we propose to integrate the latent factor model with explicit features employed in location-based recommendation.

(3) We have conducted experiments on real data sets and the results show that our hybrid method is effective, and is better than several separate methods.

2. RELATED WORK

In this section, we briefly discuss three lines of research related to recommending event-based groups: (a) group recommendation; (b) location-based recommendation; and (c) latent factor model.

Group Recommendation. In this paper, group recommendation refers to automatically recommend groups for users which they are interest to join [25, 27, 23, 5, 6]. Previous works mostly exploit latent factor models to capture the linkage information between users and groups, which could be regarded as a standard solution to group recommendation problem. Moreover, Zheng et al. [27] adopted tensor factorization to model the ternary relations among users, groups, and tags jointly in Flickr group recommendation. Chen et al. [6] extended the basic Probabilistic Latent Semantic Analysis(PLSA) model [10] to generate group content information and social relation between users and groups simultaneously. Wang et al. [25] considered both the content of image and link structure among users and groups in a joint topic model. Most of these works exploit additional information to benefit group recommendation, such as tags, content of images, etc. However, as these items or their mutual relations do not exist in our problem setting and the above models are specially designed for these information, these models cannot be directly utilized in our work.

Our event-based group recommendation problem is different from the traditional group recommendation problem, since the location information plays an important role in users' decisions about joining groups, and previous methods cannot be adapted to incorporate this information easily. Moreover, users' friendships in the real world are significant for this problem, which need to be explicitly modeled in the proposed approach.

Location-based Recommendation. Recommendations in location-based social network have been extensively studied in recent years [26, 7, 28, 19, 22]. Among them, our work is most inspired by two recent studies on extended matrix factorization [7] and supervised link prediction [22]. Cheng et al. [7] combined multi-center Gaussian model with matrix factorization for personalized point-of-interest (POI) recommendation. They drew an assumption that users' check in behaviors obey Gaussian distribution around some centers. However, this assumption is not reasonable in our problem setting since there are not enough location records for each user (many users have only several location records as they only participate in a few events) to ensure a Gaussian mixture distribution like check-in data they used. Scellato et al. [22] exploited place-based features in link prediction based on supervised learning methods. They summarized three types of features, i.e., place features, social features, and global features. Inspired by their work, we adopt some variations of those features in our work and integrate them with the latent factor model.

Latent Factor Model. There are two main types of latent factor models widely used in recommendation: matrix factorization [21, 12, 16] and topic model [2, 10, 8]. In [21, 20], a probabilistic formulation of matrix factorization has been proposed, which has spawned a large body of research. Koren at al. [12] did a series of work based on matrix factorization by considering time factor, bias influence factor, etc. In [1], Aizenberg et al. incorporated artist-enhanced latent factor into matrix factorization to help alleviate the sparsity problem in recommending music. On the other hand, topic model is similar with matrix factorization and in some situations, they can be proved equivalent [8]. Recently, Wang et.al [24] combined topic model and matrix factorization into a unified model for recommending scientific articles, which is a promising direction.

In this work, we extend basic matrix factorization model and propose a pairwise tag-enhanced latent factor model to incorporate profile information of users and groups.

3. PRELIMINARIES

3.1 Data Analysis

In this work, we take Meetup [4], an online social media site which contains various kinds of groups, as a case study. In Meetup, users can create groups, launch events in groups to attract other users to join. Or they can join other groups, participate in events organized by group members. Although these events belong to their groups uniquely, it is not necessary for users to be group members to join their events. Particularly, users can express their attitudes towards to the events by RSVP ("yes", "no" or "maybe"). Besides, users can also share some information with other users, like personal profile, comments, etc.

[4] http://www.meetup.com/

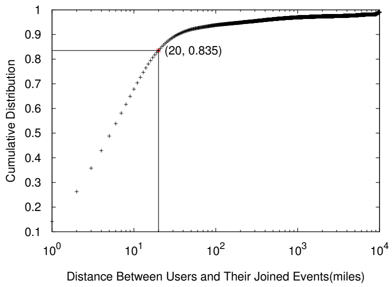


Figure 1: The Cumulative Probability Distribution of Distance Between Users and Their Joined Events

Essentially, users join this type of groups for attending some face-to-face events. Thus, the distance between users and target events is a fundamental factor that influences whether users will join groups. Via the interface of Meetup web service, users can set the range of distance from themselves to groups they will choose. Brown et al. [3] also suggested that geographical closeness could influence the formation of online communities. For further analysis, we show the cumulative probability distribution of distance between locations of users and events they have joined in Figure 1. As we can see, more than 83.5% of events are within 20 miles far from users. This is similar with the conclusions from Liu et al. [14]. They also stated the scope of locations of events that users joined in the event-based social networks is narrower than locations of check-ins in location-based social network. Usually, it happens when an event and a user are located in the same city. Besides, considering groups of all cities for recommendation is low efficient in practice. Consequently, it is more reasonable to recommend groups which usually hold events in the city where users live. In this paper, we call this problem personalized recommending event-based groups in the local city users live in. For consistent representation, we also use event-based group recommendation to denote it.

3.2 Problem Formulation

For the problem of event-based group recommendation, we have five types of entities: $\{U(\text{user}), G(\text{group}), T(\text{tag}), L(\text{location}), \text{and } E(\text{event})\}$. Given a city i, let U_i=$\{u_1, u_2, ..., u_n\}$ and G_i=$\{g_1, g_2, ..., g_m\}$ denote the set of all users and groups in the city i, respectively. For each user $u \in U$, it has a set of tags T_u to describe its preference, a unique location L_u to indicate its address, and records of events E_u and groups G_u it has joined. Similarly, for each group $g \in G$, it has also a set of tags T_g, a set of events E_g organized by its members, and a member list U_g. Besides, each event $e \in E_g$ also has a location to specify where it is held. The relations between these entities are shown in Figure 2.

Formally, the problem is defined to rank every group g belonging to the city i for a given user u who lives in the same city, according to the dyadic rating $r(u, g)$ of user u to group g, which indicates the user's preference to group g. Naturally, predicting $r(u, g)$ is the central task to be addressed here. Because users may join groups for different reasons, we should exploit the existing diverse relations between entities mentioned above to build an effective model for event-based group recommendation.

3.3 Motivating Discussion

Latent factor model is the state-of-the-art method for group recommendation problem. However, we argue that some additional key points should also be considered to solve the new problem of event-based group recommendation. In accordance with the unique characteristics of event-based groups, we make two important assumptions here.

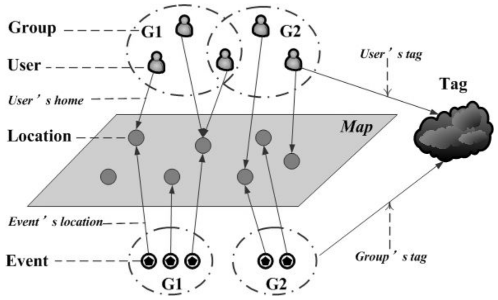


Figure 2: Overview of Entity Relations in Event-based Group Recommendation Problem

ASSUMPTION 1. *Users' friendships play a major role in users' decisions to join groups.*

This is a general assumption used in many applications of recommender system. However, we address it here because in our problem, this phenomenon is stronger. Unlike traditional online groups in which all the interactions between members only take place in a virtual network, event-based groups contain many real-world face-to-face events, which means users will meet each other when they attend the same event. Intuitively, users are more willing to meet people they are already familiar with in the real world.

ASSUMPTION 2. *Users prefer to join groups nearby.*

We have shown that users prefer to join events near their homes in Figure 1. As groups have no corresponding locations, we treat group members' locations and events' locations as an alternative choice. Formally, we assume that a group is near a user when the members in the group are near the user, or the events are held near the user. Now consider a common case that a user wants to attend a group which holds many outdoor sports like basketball game. If two groups both meet the requirements of the user but are only different in terms of distance from him, then it is more likely the user will join a nearer group due to its convenience.

In this work, we construct explicit features for predicting the dyadic rating $r(u, g)$ mainly based on the above two assumptions. Specifically, social features are extracted according to the first assumption, while location features are generated following the second assumption.

3.4 Model Framework

In order to take advantage of the latent factor model introduced in Section 2, we focus on one type of latent factor model, i.e., matrix factorization. In the setting of matrix factorization, the fundamental idea is to embody each user u and group g with low-dimension latent factors $\mathbf{p}_u$ and $\mathbf{p}_g$. Then the dyadic rating $r(u, g)$ of user u to group g is usually approximated according to the following equation,

$$\widehat{r}^m(u, g) = \mathbf{p}_u^T \mathbf{p}_g \tag{1}$$

where $\widehat{r}^m(u, g)$ denotes the rating prediction between u and g which is calculated by the method of matrix factorization.

The basic form of matrix factorization method cannot capture explicit features. However, based on our previous assumptions and analysis, explicit features play an essential role in event-based group recommendation problem. Thus, we adopt the linear model, which is widely used and effective especially when the feature space is not complex. Its basic form is shown below,

$$\widehat{r}^{l}(u,g) = \mathbf{w}^{T}\mathbf{x} \tag{2}$$

where $\mathbf{x}$ is the explicit feature vector and $\mathbf{w}$ is the coefficient vector of the linear model corresponding to each feature. Besides, $\widehat{r}^{l}(u,g)$ indicates that we adopt the linear model to get the rating prediction.

To tackle the group recommendation problem more effectively, we want to capture both implicit patterns and explicit features simultaneously. Based on this intuition, we linearly combine these methods to model both of them.

$$\widehat{r}(u,g) = \widehat{r}^{m}(u,g) + \widehat{r}^{l}(u,g) \tag{3}$$

4. EVENT-BASED GROUP RECOMMENDATION MODELING

In this section, we first introduce the explicit features in Section 4.1 and then describe the basic matrix factorization model in Section 4.2. We extend matrix factorization by proposing a pairwise tag-enhanced latent factor model in Section 4.3. Finally, we discuss two strategies for integrating matrix factorization with explicit features in Section 4.4.

4.1 Linear Model

The basic form of the linear model is shown in Equation 2. There exist more complex models which adopt complicated approaches to combine features. For example, support vector machine (SVM) can use kernel trick to nonlinearly transform features and then combine them. However, when the number of features is small, and each feature has a clear category indication, just like features built in this work, the linear model is effective enough while preserving simplicity.

Based on the two assumptions made in Section 3.3, we divide features into two categories: (1) location features; and (2) social features. We will discuss both types of the features in the following subsections in detail.

4.1.1 Location features

Location features reveal the essential distinction between event-based group recommendation and other traditional group recommendation problems.

Overlap ratio of common locations. Co-location type of features focus on the intersection part of locations both users and groups are linked to. Let $L(u)$ denote the set of locations of events user u has participated and $L(g)$ denote the set of locations of events organized by the members of group g. If u and g have many common locations, u is more likely to join g. We use Jaccard similarity to measure the overlap ratio of two location sets, i.e,

$$overlap\_location(u,g) = \frac{|L(u) \cap L(g)|}{|L(u) \cup L(g)|} \tag{4}$$

where $|L(u) \cap L(g)|$=$|L(u)| + |L(g)| - |L(u) \cup L(g)|$.

Number of common locations. As a complement to the feature above, we use the size of the intersection set instead of overlap ratio, i.e., $|L(u) \cap L(g)|$. If the size is large, user u will be more likely to join group g.

Minimum distance of users from group members. This feature measures affinity between users and groups from the perspective of spatial distance. If user u is near from a group member u' in group g, it is possible they will create links with each other and that in turn increases the possibility of u joining g.

$$user\_min\_dis(u,g) = \min_{u' \in U(g)} dis(u,u') \tag{5}$$

where $dis(u,u')$ denotes the distance between user u and u'. According to the longitude and latitude information, we use the Google Geocoding API[5] to calculate the distance.

Mean distance of users from group members. This feature is similar to the feature above except that we use mean distance instead of minimum distance to reflect the distance condition of user u from the whole group members. Thus, it is more robust which can overcome the impact of noisy users.

$$user\_mean\_dis(u,g) = \frac{\sum_{u' \in U(g)} dis(u,u')}{|U(g)|} \tag{6}$$

where $|U(g)|$ denotes the number of members in group g.

Minimum distance of users from group events. This feature directly measures the distance of user u from event e that pertains to group g. When the distance is smaller, u will be more likely to join e. That will further increase the possibility of u joining g.

$$event\_min\_dis(u,g) = \min_{e \in E(g)} dis(u,e) \tag{7}$$

where $dis(u,e)$ denotes the distance between user u and event e.

Mean distance of users from group events. Likewise, we employ average distance to provide a robust feature for measuring the spatial affinity of user u and group g.

$$event\_mean\_dis(u,g) = \frac{\sum_{e \in E(g)} dis(u,e)}{|E(g)|} \tag{8}$$

where $|E(g)|$ denotes the number of events belonging to g.

4.1.2 Social features

Social influence mainly refers to the influence of users' friends on them. Here we assume users u and u' are friends if they are online friends which means they have both joined at least one group or are offline friends which implies they have both participated in at least one common event.

Overlap ratio of common users. This feature calculates the Jaccard similarity between group g's member set and friend set of user u. We use $F(u)$ to denote the friend list of user u and $U(g)$ to represent the members of group g.

$$overlap\_user(u,g) = \frac{|F(u) \cap U(g)|}{|F(u) \cup U(g)|} \tag{9}$$

Number of common users. Similarly, we use the size of intersection set instead of overlap ratio, i.e., $|F(u) \cap U(g)|$. This feature supplements the previous feature when the group is very large so that all the overlap ratios are small.

Although social features also exploit the interaction records between users and groups like matrix factorization below, they are different because social features concentrate on direct relations while matrix factorization captures indirect relations through low dimensional representation.

[5] https://developers.google.com/maps/documentation/geocoding/

4.2 Matrix Factorization

The standard form of matrix factorization with bias terms is based on the following equation mentioned in [12],

$$\widehat{r}^{m}_{bias}(u,g) = \mu + b_u + b_g + \mathbf{p}_u^T \mathbf{p}_g \tag{10}$$

The average rating μ for user u plays a minor role because event-based group recommendation is a dyadic rating prediction problem. Moreover, user bias of u, b_u, is mainly captured by explicit features discussed before. Thus we focus on group bias here. Intuitively, more active groups will attract more users to join and this property is independent of users. In order to quantify the degree of activeness of groups, we consider two quantitative metrics here: (1) current number of members in a group; and (2) number of events organized by group members. We also adopt the linear model to determine the relative weights of these two features. Thus, we get the group bias $b_g = \mathbf{w}_g^T \mathbf{x}_g$, where $\mathbf{x}_g$ denotes the above two metrics and $\mathbf{w}_g$ represents the corresponding coefficient vector. We can consider the group bias together with social features and location features in a unified model.

4.3 Pairwise Tag-enhanced Matrix Factorization

Generally, users and groups both have profiles reflecting their preference. In this work, we only consider using tags because tags are common in many social networks, ensuring our method can be adapted to other situations. Simply matching tags based on string similarity cannot capture the semantic relatedness between tags. Thus, we also convert tags into low dimensional vectors and then multiply them to determine their relatedness. This is a powerful method for calculating words' semantic relatedness [9].

Zheng et al. [27] also considered using tags in group recommendation and adopted tensor factorization to incorporate them. More specifically, given tags users have used in labeling the pictures in groups, tensor factorization can model the ternary relationship $\langle u, g, t\rangle$. However, we cannot directly adapt it to our problem because tags are separately labeled by groups and users for themselves, which leads to two pairwise relations instead of one ternary relation.

For each user u and group g, they both have their own tags T_u and T_g, respectively. Let $\mathbf{v}_{ut}$ denote the latent factor of tag t for user u, and similarly $\mathbf{v}_{gt}$ for group g. Then we can get $\mathbf{v}_u = \frac{1}{\sqrt{|T_u|}} \sum_{t\in T_u} \mathbf{v}_{ut}$ and $\mathbf{v}_g = \frac{1}{\sqrt{|T_g|}} \sum_{t\in T_g} \mathbf{v}_{gt}$, which can be regarded as explicit profile information of users and groups. $|T_u|$ and $|T_g|$ are the number of tags of u and g, respectively. For any user or group which has no tags, we add two additional tags for them.

We incorporate pairwise tag-enhanced latent factor into matrix factorization model through directly supplementing user and group latent factors separately and then multiply the two items. The equation is defined as follows,

$$\begin{aligned}\widehat{r}^{m}_{pair}(u,g) &= (\mathbf{p}_u + \mathbf{v}_u)^T(\mathbf{p}_g + \mathbf{v}_g)\\ &= \Big(\mathbf{p}_u + \frac{1}{\sqrt{|T_u|}} \sum_{t\in T_u} \mathbf{v}_{ut}\Big)^T \Big(\mathbf{p}_g + \frac{1}{\sqrt{|T_g|}} \sum_{t\in T_g} \mathbf{v}_{gt}\Big)\end{aligned} \tag{11}$$

The above equation contains a hidden assumption that dimensions of tags and users, tags and groups should be equivalent. However, tags' optimal number of dimensions may be different from users' and groups'. In order to be more generic, we introduce a transform matrix, $\mathbf{M}$. Assume the dimension of latent variables of users and groups is m and the dimension of tags is n, then $\mathbf{M}$ is a $m \times n$ matrix. Then we get the following formula,

$$\widehat{r}^{m}_{pair}(u,g) = \Big(\mathbf{p}_u + \frac{\mathbf{M}}{\sqrt{|T_u|}} \sum_{t\in T_u} \mathbf{v}_{ut}\Big)^T \Big(\mathbf{p}_g + \frac{\mathbf{M}}{\sqrt{|T_g|}} \sum_{t\in T_g} \mathbf{v}_{gt}\Big) \tag{12}$$

When m equals n, and matrix $\mathbf{M}$ is constrained to be a diagonal matrix with all diagonal elements equal to 1, the above two equations are equivalent. Thus, Equation 11 can be regarded as a special case of Equation 12. Note that we only need to specify the dimension of matrix $\mathbf{M}$, all the elements of $\mathbf{M}$ can be learned automatically.

4.4 Model Combination

Now we integrate pairwise tag-enhanced matrix factorization with the linear model into a unified model, just as the extension of the basic framework introduced in Section 3.4. The detailed representation is shown below,

$$\begin{aligned}\widehat{r}(u,g) &= \widehat{r}^{m}_{pair}(u,g) + \widehat{r}^{l}(u,g) + b_g\\ &= \Big(\mathbf{p}_u + \frac{\mathbf{M}}{\sqrt{|T_u|}} \sum_{t\in T_u} \mathbf{v}_{ut}\Big)^T \Big(\mathbf{p}_g + \frac{\mathbf{M}}{\sqrt{|T_g|}} \sum_{t\in T_g} \mathbf{v}_{gt}\Big)\\ &\quad + \mathbf{w}_{ug}^T \mathbf{x}_{ug} + \mathbf{w}_g^T \mathbf{x}_g\end{aligned} \tag{13}$$

We call this proposed model Pairwise Tag-enhAnced and featuRe-based Matrix factorIzation for Group recommendAtioN (PTARMIGAN). In practice, we have two strategies to use training methods to learn the optimal model parameters:

- We train pairwise tag-enhanced matrix factorization and the linear model separately to get the optimal parameters for each of them. Then we combine them with different relative weights which can be determined by a grid search method.

  $$\widehat{r}(u,g) = \alpha * \widehat{r}^{m}_{pair}(u,g) + (1-\alpha) * \big(\widehat{r}^{l}(u,g) + b_g\big) \tag{14}$$

  where α is the fusion parameter to control the contribution of the two parts.
- We directly optimize Equation 13 and get all the optimal parameters simultaneously.

While the intrinsic characteristics of the two strategies are the same, we conclude two major differences between them: (1) the second strategy is sensitive to the value of the explicit features, especially when they have very large values; and (2) the first strategy needs an additional grid search method to determine the hybrid parameters. Empirically, we find that the first strategy gets a little better result in this problem, so we choose this strategy in the following.

5. MODEL TRAINING

5.1 Optimization Criterion

The central task in event-based group recommendation is to predict the dyadic rating $\widehat{r}(u,g)$ and make the groups with higher ratings rank higher. This is relevant to the ranking task in information retrieval, where the target is to rank relevant documents at top positions.

In the scenarios of information retrieval, learning to rank methods have become the state-of-the-art ranking framework [13]. Typically, learning to rank methods have been classified into three categories: pointwise approach, pairwise approach, and listwise approach. On the one hand, the pointwise approach cannot capture the relative order between training instances in the ranking list, which is significant for ranking learning. On the other hand, the training consumption of most listwise approaches is high compared with its improvement over the other two approaches. Therefore, we prefer to adopt the pairwise approach, which is a tradeoff between the pointwise and listwise approach. Finally, we choose the Bayesian Personalized Ranking (BPR) optimization criterion [18] and adapt it to our problem.

For event-based group recommendation problem, we regard the groups user u have joined as the positive group set, denoted by PI_u, while the other groups u has not joined in the same city as the negative group set, expressed as NI_u. Then the best ranking result for user u is that all groups joined by him should rank higher than other groups, which satisfies the following inequality,

$$\widehat{r}(u, g_u^p) > \widehat{r}(u, g_u^n) \qquad g_u^p \in PI_u, g_u^n \in NI_u, \forall u \in U.$$

The above inequality is relevant to the value of Area Under the ROC Curve (AUC) which is widely used in classification problem. Usually the larger the value of AUC is, the more the positive groups rank higher than the negative groups. When there are total m users, the AUC value can be calculated as follows,

$$AUC = \frac{\sum_{u=1}^{m} \sum_{g_u^p \in PI_u} \sum_{g_u^n \in NI_u} I\big(\widehat{r}(u, g_u^p) - \widehat{r}(u, g_u^n)\big)}{\sum_{u=1}^{m} |PI_u||NI_u|} \tag{15}$$

where $I(.)$ is an indicator function that is equal to 1 if $\widehat{r}(u, g_u^p) > \widehat{r}(u, g_u^n)$ and equals 0 otherwise.

Aiming for maximizing the AUC value, many works [4, 18] utilize sigmoid function, which has a form of $\sigma(x) = \frac{1}{1+e^{-x}}$, to approximate the indicator function $I(.)$ so that the objective function is derivable. Based on this trick, we can get the final target optimization function, commonly represented as a log form below,

$$\max \sum_{u=1}^{m} \sum_{g_u^p \in PI_u} \sum_{g_u^n \in NI_u} \ln\Big(\frac{1}{1 + e^{-(\widehat{r}(u, g_u^p) - \widehat{r}(u, g_u^n))}}\Big) + \text{regularization} \tag{16}$$

where the $regularization$ terms are used to avoid overfitting in the learning process. When the target is to maximize the objective function, the $regularization$ terms should take negative values. In this work, we adopt L2-regularization for model parameters Θ, i.e., $\mathbf{p}_u$, $\mathbf{p}_g$, $\mathbf{v}_{ut}$, $\mathbf{v}_{gt}$, $\mathbf{w}_{ug}$, $\mathbf{w}_g$, $\mathbf{M}$.

5.2 Parameter Learning

For both parts of the right side of Equation 14, their parameters are learned by maximizing Equation 16 using stochastic gradient descent (SGD) algorithm. SGD benefits many machine learning procedures by its fast speed to converge to global or local optimums and good scalability to large data sets. The main process of SGD is to randomly scan all training instances and iteratively update parameters. Specifically, for each training instance, we calculate its derivative and update the corresponding parameters Θ by moving along the ascending gradient direction as below,

$$\Theta \leftarrow \Theta + \eta * \frac{\partial Obj(\Theta)}{\partial \Theta} \tag{17}$$

where $Obj(.)$ denotes the objective function.

Given a training instance pair(u, g_u^p, g_u^n), we can update the related parameters by its gradient based on Equation 16,

$$\frac{\partial Obj(\Theta)}{\partial \Theta} = \epsilon * \frac{\partial \sigma\big(dif_u^{pn}\big)}{\partial \Theta} - \lambda \Theta \tag{18}$$

where $dif_u^{pn} = \widehat{r}(u, g_u^p) - \widehat{r}(u, g_u^n)$ and $\epsilon = 1 - \sigma(dif_u^{pn})$. Besides, we use λ to denote the regularization parameters.

The partial derivatives of $Obj(.)$ have the same ϵ, but different $\frac{\partial \sigma(dif_u^{pn})}{\partial \Theta}$. According to Equation 13 and Equation 14, the detailed gradients of the corresponding latent variables in matrix factorization can be derived as follows,

$$\frac{\partial \sigma(dif_u^{pn})}{\partial \mathbf{p}_u} = \Big(\mathbf{p}_g^p + \frac{\mathbf{M}}{\sqrt{|T_g^p|}} \sum_{t \in T_g} \mathbf{v}_{gt}^p\Big) - \Big(\mathbf{p}_g^n + \frac{\mathbf{M}}{\sqrt{|T_g^n|}} \sum_{t \in T_g} \mathbf{v}_{gt}^n\Big) \tag{19}$$

$$\frac{\partial \sigma(dif_u^{pn})}{\partial \mathbf{p}_g^p} = \Big(\mathbf{p}_u + \frac{\mathbf{M}}{\sqrt{|T_u|}} \sum_{t \in T_u} \mathbf{v}_{ut}\Big) \tag{20}$$

$$\frac{\partial \sigma(dif_u^{pn})}{\partial \mathbf{v}_{gt}^p} = \frac{\mathbf{M}^T}{\sqrt{|T_g^p|}} \Big(\mathbf{p}_u + \frac{\mathbf{M}}{\sqrt{|T_u|}} \sum_{t \in T_u} \mathbf{v}_{ut}\Big) \tag{21}$$

We only list a part of gradients of the latent variables for saving space. However, all the other latent variables have similar gradient formulas. The gradients for matrix $\mathbf{M}$ can be induced as below,

$$\begin{aligned}\frac{\partial \sigma(dif_u^{pn})}{\partial \mathbf{M}_{[i,j]}} = &\Big(\frac{1}{\sqrt{|T_u|}} \sum_{t \in T_u} \mathbf{v}_{ut,[j]}\Big)\Big(\mathbf{p}_{g,[i]}^p + \frac{\mathbf{M}_{[i,*]}}{\sqrt{|T_g^p|}} \sum_{t \in T_g^p} \mathbf{v}_{gt}^p\Big) \\ &+ \Big(\frac{1}{\sqrt{|T_g^p|}} \sum_{t \in T_g^p} \mathbf{v}_{gt,[j]}^p\Big)\Big(\mathbf{p}_{u,[i]} + \frac{\mathbf{M}_{[i,*]}}{\sqrt{|T_u|}} \sum_{t \in T_u} \mathbf{v}_{ut}\Big) \\ &- \Big(\frac{1}{\sqrt{|T_u|}} \sum_{t \in T_u} \mathbf{v}_{ut,[j]}\Big)\Big(\mathbf{p}_{g,[i]}^n + \frac{\mathbf{M}_{[i,*]}}{\sqrt{|T_g^n|}} \sum_{t \in T_g^n} \mathbf{v}_{gt}^n\Big) \\ &- \Big(\frac{1}{\sqrt{|T_g^n|}} \sum_{t \in T_g^n} \mathbf{v}_{gt,[j]}^n\Big)\Big(\mathbf{p}_{u,[i]} + \frac{\mathbf{M}_{[i,*]}}{\sqrt{|T_u|}} \sum_{t \in T_u} \mathbf{v}_{ut}\Big)\end{aligned} \tag{22}$$

where $\mathbf{M}_{[i,j]}$ denotes the $[i, j]$th element in $\mathbf{M}$, $\mathbf{M}_{[i,*]}$ denotes the ith row in $\mathbf{M}$, and $\mathbf{v}_{ut,[j]}$ denotes the jth element in the latent vector. For explicit features in the linear model, the gradients of their coefficients are shown as follows,

$$\frac{\partial \sigma(dif_u^{pn})}{\partial \mathbf{w}} = \mathbf{x}^p - \mathbf{x}^n \tag{23}$$

Finally, each model parameter should add one regularization term as Equation 18 shows. Since tags (V_{ut}, V_{gt}^p, V_{gt}^n) may have overlaps, we should add the regularization terms until the end of each iteration.

The learning process continues until one of the following two termination conditions is satisfied: (1) the value of AUC on training data set remains stable in consecutive iterations; and (2) the maximum number of iterations is reached. Generally, this number can be set empirically.

6. EXPERIMENTAL STUDY

6.1 Datasets Allocation

As stated in Section 3.1, we address the problem of event-based group recommendation for users in the same city. We selected New York City(NYC)[6] and Los Angeles(LA)[7] to analyze because they are among the largest cities in USA and hence have more users and groups than many other cities. Similarly, [19, 11] also choose the two cities in their experiments on location-based recommendation. We got the data set for the two cities in Meetup by extracting them from the data set published by [14].

Since all locations of users and events are represented by longitude and latitude, without explicitly specifying their cities. We first got users within a distance range from the centers of the two cities. Then starting from these users, we extracted groups that hold events within a distance range

[6] http://en.wikipedia.org/wiki/New_York_City
[7] http://en.wikipedia.org/wiki/Los_Angeles

from these users. Besides, we extracted their corresponding tags, events, and locations. To remove noisy data and guarantee the reliability of the experimental results, we further selected users who have joined at least 5 groups and groups which have at least 5 members. We also regarded the users who have joined most of the groups in the city as noisy users and removed them. For tags, we split long tag string and retained tags frequently used by users and groups. Finally, we obtained the experimental datasets. The details of the datasets are shown in Table 1.

City	User	Group	Tag	Event	Location
NYC	5001	2494	2137	95339	16963
LA	10944	2783	3421	120310	21389

Table 1: Statistics of Event-based Group Data

As we adopt the BPR framework for learning parameters, we need to construct training data including a positive and a negative group in each training instance. However, if we use all groups (except those user u has joined) in a city as negative groups to build training pairs for u, then the total number of pairs for all users is huge which will lead to a very slow training scheme. For simplicity, we adopted an alternative strategy by randomly sampling 10 groups users have not joined for each positive group to constitute the training pairs. Therefore, the ratio of the number of positive groups to negative groups is 1:10. For both NYC and LA data sets, we randomly split them with 70% into the training sets, 10% into the validation sets, and 20% into the test sets. Because the test data sets are composed of group pairs, we call them **pair test sets**. These test sets are mainly employed for comparing the performance of different methods.

In order to evaluate the effectiveness of event-based group recommendation in a real scenario, i.e., using all groups in a city as candidates, we first randomly sampled 1000 users as test users for both cities. Then for each user, we removed the groups he has joined and also used in the training and validation sets. For the remaining groups he has joined, we regard them as the target groups and hope the proposed method to give them higher ratings than other remaining groups in the city. Finally, we get the two cities' test sets which we call **list test sets** because the test results are evaluated on the group ranking lists.

6.2 Evaluation Measures

To evaluate the group recommendation results, we adopt three standard evaluation metrics: AUC, P@k (Precision at Position k), and MAP (Mean Average Precision) .

AUC measures the overall results of classification. It is suitable for highly imbalanced data set, as in our case where the negative groups constitute a high proportion. In this work, we use AUC in the pair test sets to measure the results.

P@k and MAP are mainly used in ranking problems. For each user u, suppose positive groups correspond to the category C_T, then average precision (AP) is defined as follows,

$$AP = \frac{\sum_{k=1}^{n} P@k \times I\big(\pi(k, L(u)) = C_T\big)}{|G_u|} \tag{24}$$

where n is the number of groups, $\pi(i, L(u))$ denotes the category of the group at position i in the ranking list $L(u)$ and G_u represents the groups joined by u in the test sets. Finally, we can obtain MAP by averaging AP for all users. In this paper, we use P@k and MAP in the list test sets.

6.3 Baseline Methods

Since the proposed method, PTARMIGAN, is a combination of the latent factor model and the linear model with explicit features, we focus on demonstrating that the hybrid method is suitable for the new problem of event-based group recommendation and it is better than any single method. Besides, we want to verify the effectiveness of each type of features. With these goals in mind, the baselines we adopt in this work are listed as follows,

- **Linear model with group activeness features (LiGF)**: This method only considers group activeness features. Therefore, it ignores personal preference of users.
- **Linear model with location features (LiPF)**: This method corresponds to Assumption 1. It reflects the affinity between users and groups from the spatial perspective.
- **Linear model with social features (LiSF)**: This method corresponds to Assumption 2. It is used to measure how users are influenced by their friends when deciding whether to join groups.
- **Linear model with all explicit features (LiAF)**: This method integrates all explicit features we used. It can be viewed as a special case of our proposed method when only considering explicit features. Compared with previous three baselines, it can be used to verify the advantage of integrating all types of features.
- **Matrix factorization (MF)**: It is a fundamental type of latent factor model for recommending groups discussed in Section 4.2. Many methods for recommending groups are variants of this model. However, as stated in Section 2, these methods use additional information like content of images and cannot be directly utilized in our problem. Thus, the basic latent factor models are used here as baselines to show the results of recommending event-based groups. Besides MF, we also tried other two standard latent factor models, PLSA and LDA, in our problem. However, MF with the BPR learning framework outperforms these two methods. Due to space limitation, we only analyze the results of MF in our experimental study.
- **Pairwise tag-enhanced matrix factorization (P-TMF)**: This method is an extension of matrix factorization by incorporating pairwise tag latent factor to enhance factors of users and groups. It can be regarded as considering additional content information in group recommendation.

6.4 Parameter Setup

Learning rate and regularization parameters. Learning rate controls the speed of model training. However, it may not be able to converge if it is set too large. In this work, the learning rate is set to 0.01 for matrix factorization and 10^{-4} for the linear model. On the other hand, regularization parameters are empirically set to 0.1 for matrix factorization and 10^{-3} for the linear model.

Relative weight $\boldsymbol{\alpha}$. α is the fusion coefficient of Equation 14. We tune α by evaluating how AUC changes in validation sets. As the results shown in Figure 3, we get stable and better performance when $\alpha \in [0.4, 0.9]$. Taking the results of P@k and MAP into consideration, we set $\alpha = 0.85$.

Algorithm	LiGF	LiPF	LiSF	LiAF	MF	PTMF	**PTARMIGAN**
NYC	0.716	0.740	0.801	0.867	0.872	0.874	**0.885**
LA	0.793	0.805	0.861	0.894	0.899	0.901	**0.912**

Table 2: AUC of pair test set

Dimension of latent factors. In addition to the above parameters, we also conducted sensitivity analysis in terms of the dimensionality of the latent factors. As we varied the number of dimensions, we found that it is not very sensitive. Empirically, we set the number of dimensions to be 20 for the latent factors in our model .

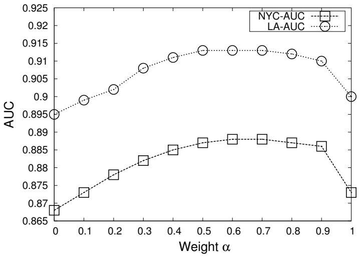


Figure 3: The Effect of Weight α

6.5 Experimental Results

(A) Results on Pair Test Sets. We begin with reporting the results on pair test sets. We adopt AUC metric because it can reveal the overall results of all methods under the adopted pairwise learning framework, i.e., BPR .

All the results are shown in Table 2. Now we first discuss the contribution of different types of explicit features by comparing LiGF, LiPF, and LiSF. Among the three baselines, the performance of LiGF is worse than the other two methods because it ignores personal factors which play a significant role in personalized recommendation. However, the results also reflect users have a tendency to join active groups. For LiPF and LiSF, the latter performs better than the former method. It corresponds to our intuition that user's friends have a major influence on user's decisions to join groups, especially for face-to-face events. Moreover, we find the combination of these explicit features obtains better results, with 8.25% improvement in NYC and 3.71% improvement in LA pair test sets over the best sub-method. This demonstrates different types of features can be complementary to each other from different aspects of the problem.

Then we analyze the difference between MF and PTMF. As we discussed in Section 4.3, PTMF is an extension of MF by considering additional tag information. As the results show, PTMF behaves a little better than MF. This verifies incorporating profile information of users and groups indeed somewhat captures their characteristic and preference. In fact, when a user and a group have similar profile information, it is reasonable they are more likely to create links. However, more than 20% users in both of our pair data sets have no tags which may lead to that the improvements here are not so obvious as we expected.

Finally, we compare the proposed method, PTARMIGAN, with the methods, LiAF and MF. By fusing explicit features and latent factors into a unified model, AUC of PTARMIGAN is further improved over the two strong baselines in both test sets. This demonstrates that the combination of matrix factorization and explicit features is effective for event-based group recommendation. We notice that AUC of all methods on LA pair test set are superior to the results on NYC pair test set. This can be explained by the fact that NYC data set is sparser than LA data set.

(B) Results on List Test Sets. We concentrate on the top-3 precision of recommendation results when all groups in a city are considered because of the following reasons. First, in the test data, many users have only one positive instance. Thus, it is fair to compare all methods on the top recommendation results. Second, in a real recommendation scenario, most users only join a few groups, and users tend to only focus on the top recommendation results while ignoring the rest. We also utilize MAP to measure the overall results of recommendation.

Since LiAF always performs better than the methods using incomplete explicit features, we only use it among the feature-based methods for comparison in this test. As Figure 4 shows, PTARMIGAN achieves the best results in P@1, P@3 and MAP. For the top-1 precision, it achieves 7.25% and 8.11% relative improvements in NYC and LA list test sets respectively over the second best corresponding results. It also achieves the best MAP among all methods. Thus, it verifies that the proposed method can indeed achieve better group recommendation results.

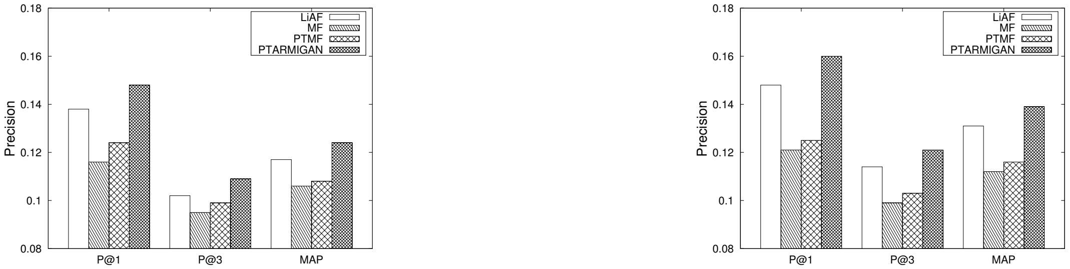
An interesting phenomenon we found is that LiAF performs a little better than both MF and PTMF in this case, which is contrary to the results on the pair test sets. This may be caused by the following two reasons: (1) explicit features, especially social and place features are significant in the problem of event-based group recommendation; and (2) pairwise ranking learning strategy may lose some accuracy when the number of relevant documents is much smaller than irrelevant documents, just like group ranking here, which can also be regarded as an extremely skewed binary classification problem.

To sum up, our proposed method, PTARMIGAN, achieves the best results among the above methods. Now, we further discuss the complexity of our method and the cold start problem in recommendation.

Parameter complexity. The number of parameters in the linear model is determined by the number of types of features, which is expressed as L. In the part of the latent factor model, the complexity of the parameters is $O(K_{ui}N_u + K_{ui}N_i + K_tN_t + K_{ui}K_t)$, where K_{ui} and K_t correspond to the dimension of users/items and tags respectively, N_u denotes the number of users and it is similar for N_i and N_t. Thus, the parameter complexity of our hybrid method is $O(K_{ui}N_u + K_{ui}N_i + K_tN_t + KK_t + L)$, which grows linearly with the size of data.

Time complexity. Suppose we need R_1 and R_2 iterations to train the two parts of our model respectively and c is the time cost of updating one parameter, then the training time complexity is $O(cR_1 * (K_{ui}N_u + K_{ui}N_i + K_tN_t + KK_t) + cR_2 * L)$. Usually R_1, R_2, and c are small in practice. Thus, the training time also grows linearly with the size of data.

Cold start problem. Cold start problem is the main challenge in recommendation. For both cold start users and groups, we have none of their interaction history and the basic latent factor model cannot be applied to this situa-



(a) P@n and MAP of NY list test set

(b) P@n and MAP of LA list test set

Figure 4: Results of compared methods on list test set

tion. While we do not emphasize this in this work, the proposed method should be better at handling it than the basic latent factor models, due to its ability of flexible demographic feature integration, such as locations and tags used in this paper. Normally, these features can be easily mined from the cold start users and groups as they are often required to fill out their profiles when they create accounts.

7. CONCLUSION

In this paper, we study a new type of group recommendation problem: personalized recommending event-based groups to users. In order to model the essential properties of event-based groups, we first adopt the linear model to capture explicit features, including location features and social features. Then, we utilize matrix factorization, one type of the latent factor model which is widely used in group recommendation problem, to model past interactions between users and groups. We also consider the group bias in matrix factorization represented in terms of group activeness-based features. Moreover, we extend the basic matrix factorization method by incorporating pairwise tag-enhanced latent factor. Through this way, profile information of users and groups is also considered. Finally, our proposed method, PTARMIGAN, integrates all the above elements into a unified model. We conducted experiments on data sets from a popular website, Meetup, and the results demonstrate the effectiveness of our method.

8. ACKNOWLEDGMENTS

This work was supported in part by National Natural Science Foundation of China under Grant No. 61272088 and National Basic Research Program of China (973 Program) under Grant No. 2011CB302206.

9. REFERENCES

[1] N. Aizenberg, Y. Koren, and O. Somekh. Build your own music recommender by modeling internet radio streams. In *WWW*, pages 1–10, 2012.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[3] C. Brown, V. Nicosia, S. Scellato, A. Noulas, and C. Mascolo. Where online friends meet: Social communities in location-based networks. In *ICWSM*, 2012.

[4] K. Chen, T. Chen, G. Zheng, O. Jin, E. Yao, and Y. Yu. Collaborative personalized tweet recommendation. In *SIGIR*, pages 661–670, 2012.

[5] W.-Y. Chen, J.-C. Chu, J. Luan, H. Bai, Y. Wang, and E. Y. Chang. Collaborative filtering for orkut communities: discovery of user latent behavior. In *WWW*, pages 681–690, 2009.

[6] W.-Y. Chen, D. Zhang, and E. Y. Chang. Combinational collaborative filtering for personalized community recommendation. In *KDD*, pages 115–123, 2008.

[7] C. Cheng, H. Yang, I. King, and M. R. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *AAAI*, 2012.

[8] C. Ding, T. Li, and W. Peng. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics & Data Analysis*, 52(8):3913–3927, 2008.

[9] G. Halawi, G. Dror, E. Gabrilovich, and Y. Koren. Large-scale learning of word relatedness with constraints. In *KDD*, pages 1406–1414, 2012.

[10] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, pages 50–57, 1999.

[11] Y. Z. Jie Bao and M. F.Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *ACM GIS*, 2012.

[12] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug. 2009.

[13] T.-Y. Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331, Mar. 2009.

[14] X. Liu, Q. He, Y. Tian, W.-C. Lee, J. McPherson, and J. Han. Event-based social networks: linking the online and offline social worlds. In *KDD*, pages 1032–1040, 2012.

[15] X. Liu, Y. Tian, M. Ye, and W.-C. Lee. Exploring personal impact for group recommendation. In *CIKM*, pages 674–683, 2012.

[16] S. Moghaddam, M. Jamali, and M. Ester. Etf: extended tensor factorization model for personalizing prediction of review helpfulness. In *WSDM*, pages 163–172, 2012.

[17] J. A. Recio-Garcia, G. Jimenez-Diaz, A. A. Sanchez-Ruiz, and B. Diaz-Agudo. Personality aware recommendations to groups. In *RecSys*, pages 325–328, 2009.

[18] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, Arlington, Virginia, United States, 2009.

[19] A. Sadilek, H. Kautz, and J. P. Bigham. Finding your friends and following them to where you are. In *WSDM*, pages 723–732, 2012.

[20] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the International Conference on Machine Learning*, volume 25, 2008.

[21] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

[22] S. Scellato, A. Noulas, and C. Mascolo. Exploiting place features in link prediction on location-based social networks. In *KDD*, pages 1046–1054, 2011.

[23] V. Vasuki, N. Natarajan, Z. Lu, and I. S. Dhillon. Affiliation recommendation using auxiliary networks. In *Recsys*, pages 103–110, 2010.

[24] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *KDD*, pages 448–456, 2011.

[25] J. Wang, Z. Zhao, J. Zhou, H. Wang, B. Cui, and G. Qi. Recommending flickr groups with social topic model. *Inf. Retr.*, 15(3-4):278–295, June 2012.

[26] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*, pages 325–334, 2011.

[27] N. Zheng, Q. Li, S. Liao, and L. Zhang. Flickr group recommendation based on tensor decomposition. In *SIGIR*, pages 737–738, 2010.

[28] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with gps history data. In *WWW*, pages 1029–1038, 2010.