Followee Recommendation in Event-Based Social Networks

Shuchen Li, Xiang Cheng(✉), Sen Su, and Le Jiang

State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications, Beijing, China
{lsc,chengxiang,susen,jiangle}@bupt.edu.cn

Abstract. Recent years have witnessed the rapid growth of event-based social networks (EBSNs) such as Plancast and DoubanEvent. In these EBSNs, followee recommendation which recommends new users to follow can bring great benefits to both users and service providers. In this paper, we focus on the problem of followee recommendation in EBSNs. However, the sparsity and imbalance of the social relations in EBSNs make this problem very challenging. Therefore, by exploiting the heterogeneous nature of EBSNs, we propose a new method called Heterogenous Network based Followee Recommendation (HNFR) for our problem. In the HNFR method, to relieve the problem of data sparsity, we combine the explicit and latent features captured from both the online social network and the offline event participation network of an EBSN. Moreover, to overcome the problem of data imbalance, we propose a Bayesian optimization framework which adopts pairwise user preference on both the social relations and the events, and aims to optimize the area under ROC curve (AUC). The experiments on real-world data demonstrate the effectiveness of our method.

Keywords: Followee recommendation · Event-based social networks · Heterogenous network

1 Introduction

In the past few years, event-based social networks (EBSNs), such as Plancast[1] and DoubanEvent[2], have grown rapidly and attracted millions of users. These EBSNs provide online platforms for users to establish, manage and join social events. In these EBSNs, followee recommendation can bring great benefits to both users and service providers. On one hand, users could find like-minded people or their friends in real life to follow, and thus are able to enjoy better user experiences through effective followee recommendation. On the other hand, service providers can exploit followee recommendation to drive users' engagement and loyalty. In this paper, we focus on the problem of followee recommendation in EBSNs.

[1] http://www.plancast.com/.
[2] http://beijing.douban.com/.

H. Gao et al. (Eds.): DASFAA 2016 Workshops, LNCS 9645, pp. 27–42, 2016.
DOI: 10.1007/978-3-319-32055-7_3

To make a study, we collect real data from Plancast and DoubanEvent. Based on our analysis, the social relations in Plancast and DoubanEvent are extremely sparse and imbalanced. In particular, the distributions of the number of followees and followers in both the datasets almost follow a power-law distribution. The problem of data sparsity and imbalance can significantly degrade the recommendation performance, due to the lack of enough data and the huge amount of negative samples. However, since an EBSN is a heterogeneous network which consists of an online social network and an offline event participation network, in addition to the social relations, there is an unprecedented source which can be utilized for our problem: the event participation records. As users attend events mainly based on their interests, the events attended by a user can reflect the user's interest. According to the social theory of homophily, users with similar interests are more likely to establish social relations. Besides, since the events are held at physical places, users who have attended the same event may have a chance to meet each other and develop new social links between them. Therefore, we can exploit the event participation records to improve the effectiveness of followee recommendation.

To relieve the problem of data sparsity, we utilize both the social relations and event participation records for followee recommendation. In particular, we extract two kinds of explicit features for our problem: (1) social features, which are extracted from the online social network and (2) event-based features, which are captured from the offline event participation network. Moreover, to take advantage of the latent factor model, we employ matrix factorization model in the online social network and the offline event participation network to capture the latent features in both the networks. More importantly, to derive users's latent features better and more comprehensively, we assume that these two networks share the same latent user feature in the matrix factorization model for each user. Finally, we combine all the explicit and latent features into a unified recommendation model.

To overcome the problem of data imbalance, a common way is to consider the area under ROC curve (AUC) as the optimization object, which is not influenced by the distribution of classes. Since our problem can be regarded as a ranking problem with implicit feedback, inspired by the Bayesian Personal Ranking (BPR) [16], we propose a Bayesian optimization framework which aims to optimize the AUC. By exploiting the heterogeneous network, our framework adopts pairwise user preference on both the social relations and the events, and optimizes for ranking pairs correctly.

To sum up, the primary contributions of our research are listed as follows.

- To the best of our knowledge, we are the first to study the problem of followee recommendation in EBSNs.
- We extract several social and event-based explicit features from the online social network and the offline event participation network. Moreover, we combine all the explicit and latent features into a unified recommendation model.
- We propose a new Bayesian optimization framework which adopts pairwise user preference on both the social relations and the events.

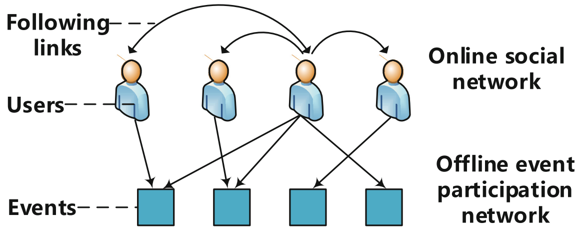


Fig. 1. An illustration of EBSNs.

– We evaluate the performance of our method using real-world data collected from Plancast and DoubanEvent. Experimental results show that our method is superior to alternatives and methods that consider only part of the factors exploited in this paper.

The rest of this paper is organized as follows. In Sect. 2, we give an overview of EBSNs, formally define the problem, and make some analyses about our data. In Sect. 3, we show the details of our followee recommendation model. We report the experimental results in Sect. 4 and review related works in Sect. 5. Finally, we make the conclusion in Sect. 6.

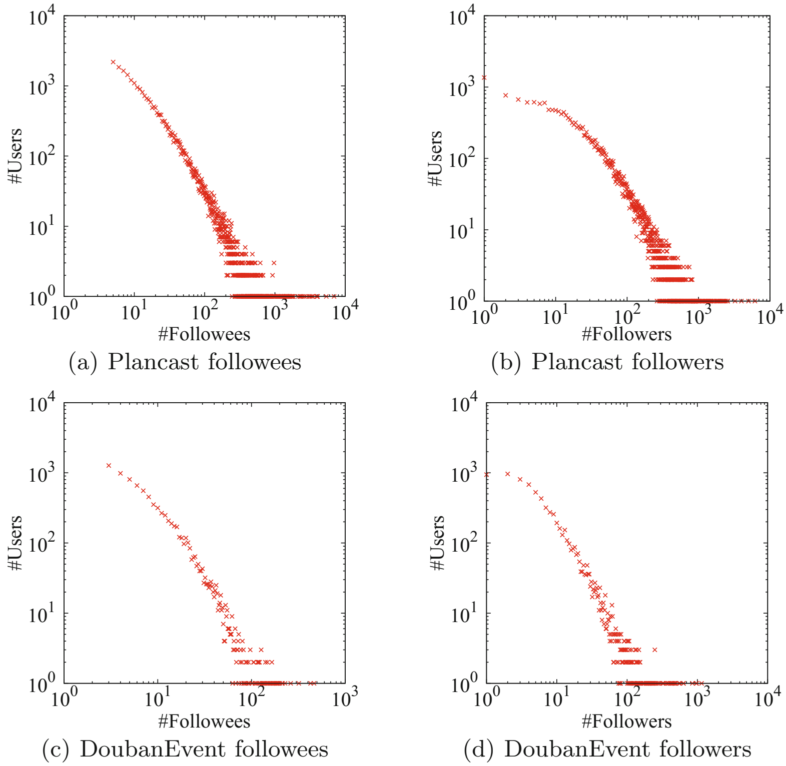
2 Preliminaries

2.1 Event-Based Social Network

A graph representation of EBSNs such as Plancast and DoubanEvent is shown in Fig. 1. We can observe that users and events are the two main entities in an EBSN. In particular, an event is an activity that is held in a physical venue, e.g., a drama held in a theater. Moreover, users are the participants of events, who can express their willingness to join an event by RSVP ('Yes' or 'Maybe'). The RSVP('Yes' or 'Maybe') indicates that a user wants to attend or is interested in an event. Besides, they can establish social relations by following other users. From Fig. 1, we can also find that the network structure of an EBSN is heterogeneous, which consists of an online social network and an offline event participation network.

2.2 Problem Definition

In an EBSN, we have two types of entities: $\{U(\text{user})$ and $E(\text{event})\}$, and two kinds of networks: $\{G^{on}(\text{online social network})$ and $G^{off}(\text{offline event participation network})\}$. Let $U = \{u_1, u_2, ..., u_n\}$ denote the set of users and $E = \{e_1, e_2, ..., e_m\}$ denote the set of events, respectively. For each user $u \in U$, it has a set of followees $\mathcal{F}_u^+ \subseteq U$, a set of followers $\mathcal{F}_u^- \subseteq U$, and a set of attended events $E_u \subseteq E$. For convenience, we use a matrix $R \in \mathbb{R}^{n \times n}$ to represent the online social network G^{on}, where $r_{ij} = 1$ indicates that user u_i is a follower of user u_j and $r_{ij} = 0$ means



(a) Plancast followees (b) Plancast followers

(c) DoubanEvent followees (d) DoubanEvent followers

Fig. 2. Followee and Follower distributions

that user u_i has not yet followed user u_j. Similarly, let matrix $V \in \mathbb{R}^{n \times m}$ represent the offline event participation network G^{off}, where v_{ij} is equal to 1 if user u_i has attended event e_j and equal to 0 otherwise. Then, the problem of followee recommendation can be formally defined as ranking all users in the candidate set $C_u = U - (\mathcal{F}_u^+ \cup u)$ for each user $u \in U$, according to user u's preference to other users, and recommending top-k users for user u.

2.3 Data Analysis

To make a study of the problem, we collect real data from Plancast and DoubanEvent. The Plancast dataset was published by the work [13] and the DoubanEvent dataset was obtained by querying Douban API[3]. To make data sufficient for evaluation, for the Plancast dataset, we remove the inactive users who have less than 5 followees and have not attended any event; for the DoubanEvent dataset, we remove the inactive users who have less than 3 followees and have not joined any event. After preprocessing, some statistics of the two datasets

[3] http://developers.douban.com/

Table 1. Statistics of the pre-processed datasets

	Plancast	DoubanEvent
Number of users	28,060	8,299
Number of events	237,994	104,591
Number of follower-followee links	1,228,714	103,604
Number of use-event links	591,870	531,113
Max number of followees	6,943	465
Max number of followers	6,149	1,148

are shown in Table 1. In summary, we obtain the density of the online social network (i.e., the social relations) as 1.56×10^{-3} for Plancast dataset and 1.5×10^{-3} for DoubanEvent dataset, which indicates the high sparsity of the social relations.

To further understand the social relations in Plancast and DoubanEvent, we analyze the distributions of the followee and follower numbers in both the datasets. As shown in Fig. 2, the distributions in both datasets almost follow a power-law distribution, which means that while most of users have a small amount of followees and followers, there exist a few users who have a large number of followees and followers. On average, each user of Plancast has 44 followees and a user in DoubanEvent has 12 followees, which indicates the extreme imbalance of the social relations, i.e., the number of a user's followees is very small with regard to the total number of the users.

To sum up, the social relations in both the datasets are extremely sparse and imbalanced, which makes the problem of followee recommendation very challenging.

3 Followee Recommendation Modeling

In this section, we introduce the details of our model. We first present the explicit features and the latent features in Sects. 3.1 and 3.2, respectively. Then, we introduce our recommendation model in Sect. 3.3. Finally, we describe the parameter learning method in Sect. 3.4.

3.1 Explicit Features

Explicit features for a pair of users are usually used to measure the similarity of two users from different points of view. According to the social theory of homophily, two users are likely to develop new social links between them if they are similar to each other. In this work, we generate two kinds of explicit features: social features and event-based features. In the following, we describe each explicit feature in detail.

Social Features. Online social relations intuitively play an important role in helping create new social links. In the following, we define several social features based on node neighborhoods in the online social network of the EBSN.

Number of common neighborhoods. Given a user pair $\langle u_i, u_j \rangle$, this feature calculates how many user u_i's followees have followed user u_j and is defined as follows:

$$common\_neighbor(u_i, u_j) = \left|\mathcal{F}_i^+ \cap \mathcal{F}_j^-\right|.$$

Ratio of overlapped neighborhoods. This feature measures the Jaccard similarity between user u_i's followees and user u_j's followers, which is defined as follows:

$$overlap\_neighbor(u_i, u_j) = \frac{\left|\mathcal{F}_i^+ \cap \mathcal{F}_j^-\right|}{\left|\mathcal{F}_i^+ \cup \mathcal{F}_j^-\right|}.$$

Adamic/Adar. This feature measures the Adaminc/Adar [1] score of two users, which sums up the reciprocal value of a logarithmic function with the number of each common neighbor's followees, and is defined as follows:

$$aa\_neighbor(u_i, u_j) = \sum_{c \in \mathcal{F}_i^+ \cap \mathcal{F}_j^-} \frac{1}{\log(|\mathcal{F}_c^+|)}.$$

Event-Based Features. Offline event participation is a unique characteristic of EBSNs, compared with conventional social networks. In this work, we define several event-based features for a user pair $\langle u_i, u_j \rangle$ as follows.

Number of common events. This feature captures the number of common events attended by two users and is formally defined as follows:

$$common\_event(u_i, u_j) = |E_i \cap E_j|.$$

Ratio of overlapped events. This feature measures the overlap ratio of two user's event sets using the Jaccard similarity, which is defined as follows:

$$overlap\_event(u_i, u_j) = \frac{|E_i \cap E_j|}{|E_i \cup E_j|}.$$

Adamic/Adar with event entropy. Distinct events may have different impacts on social link creation and, intuitively, events with a few attendees have a larger probability of creating new social links among the attendees than those with a large number of attendees, e.g., the attendees of a small home party are more likely to be good friends, while large events are usually public events such as concerts and exhibitions. To estimate the weights of the events, we introduce an entropy-based measure based on the information theory. Let N_{e_k} denote the number of attendees of event e_k and we use a discrete uniform distribution

$p_i = 1/N_{\mathrm{e}_k}$ to describe the proportion of a certain user among the attendees. Let EN_{e_k} denote the entropy of event e_k, then we define EN_{e_k} as follows:

$$EN_{e_k} = -\sum_i p_i \log p_i = \log N_{\mathrm{e}_k}.$$

As shown, the event entropy has a positive correlation with the number of attendees of an event. Inspired by the Adamic/Adar measure, we define a feature *ent_event* as the sum of the reciprocal value of each event entropy in common events between two users, i.e.,

$$en\_event(u_i, u_j) = \sum_{e_k \in E_i \cap E_j} \frac{1}{EN_{e_k}}.$$

Weighted common events. This feature takes the weights of events into consideration. In particular, we assign an event participation vector $\mu_i = (\mu_{i1}, \mu_{i2}, ..., \mu_{im})$ for each user $u_i \in U$, where if the user has attended event e_j then μ_{ij} is the reciprocal value of the entropy of event e_j, i.e., $\mu_{ij}{=}1/EN_{e_j}$, otherwise, $\mu_{ij} = 0$. Then, we define a feature *w_common_event* as the inner product of two users' event participation vectors, i.e.,

$$w\_common\_event(u_i, u_j) = \mu_i \cdot \mu_j.$$

Weighted overlapped events. Similar to weighted common events, this feature also takes the weights of events into consideration but focuses on the overlapped events. Given two users, we define a feature *w_overlap_event* as the cosine similarity of their event participation vectors, i.e.,

$$w\_overlap\_event(u_i, u_j) = \frac{\mu_i \cdot \mu_j}{\|\mu_i\| \times \|\mu_j\|}.$$

Since each aforementioned explicit feature has a clear meaning and the feature space is small, to derive a user's preference towards another user based on the explicit features, we adopt the linear model to combine all the explicit features, which is simple but effective enough. Let z_{ij} denote the explicit feature vector of the user pair $\langle u_i, u_j \rangle$ and $r_f^u(u_i, u_j)$ denote the user u_i's preference towards user u_j based on the explicit features. Then, we define $r_f^u(u_i, u_j)$ as follows:

$$r_f^u(u_i, u_j) = w^T z_{ij}, \tag{1}$$

where w is the weight coefficient vector.

3.2 Latent Features

In this work, we employ matrix factorization model to capture the latent features of users and events, which is widely employed in recommendation system. The main idea of matrix factorization model is to seek a low-dimension latent feature vector to represent each entity and the rating scores can be approximated by a function of these low-dimension feature vectors. In our problem, there are two kinds of entities: users and events, correspondingly, we use latent feature vectors $x_i \in \mathbb{R}^{d\times 1}$ and $y_j \in \mathbb{R}^{d\times 1}$ with $d \ll |U| \wedge d \ll |E|$ to represent each user $u_i \in U$ and each event $e_j \in E$. Let $r_m^u(u_i, u_j)$ and $r_m^e(u_i, e_k)$ denote the user u_i's preference towards user u_j and event e_k based on the latent features, respectively. Then, we can define $r_m^u(u_i, u_j)$ and $r_m^e(u_i, e_k)$ as follows:

$$\begin{aligned} r_m^u(u_i, u_j) &= bu_i + bu_j + {x_i}^T H x_j, \\ r_m^e(u_i, e_k) &= bu_i + be_k + {x_i}^T y_k, \end{aligned} \tag{2}$$

where the matrix $H \in \mathbb{R}^{d\times d}$ represents the correlations among users' latent features. Besides, bu_i, bu_j, and be_k are the bias items of user u_i, user u_j, and event e_k, respectively. Notice that, these two preferences share the same latent user feature for each user.

3.3 Recommendation Model

In this work, we combine the explicit and latent features to obtain a user's overall preference towards another user. Let $r^u(u_i, u_j)$ denote the user u_i's overall preference towards user u_j. Then, we define $r^u(u_i, u_j)$ as follows:

$$\begin{aligned} r^u(u_i, u_j) &= r_f^u(u_i, u_j) + r_m^u(u_i, u_j) \\ &= w^T z_{ij} + bu_i + bu_j + {x_i}^T H x_j. \end{aligned} \tag{3}$$

Moreover, we derive a user's overall preference towards an event only based on the latent features. Let $r^e(u_i, e_k)$ denote the user u_i's overall preference towards event e_j. Then, we define $r^e(u_i, e_k)$ as follows:

$$r^e(u_i, e_k) = r_m^e(u_i, e_k) \ = \ bu_i + be_k + {x_i}^T y_k. \tag{4}$$

Since our problem can be regarded as a ranking problem with implicit feedback, inspired by the BPR, we propose a Bayesian optimization framework which aims to optimize the AUC. In particular, for each user $u_i \in U$, it has two user sets: one is his/her followees (i.e., the positive user set), denoted as P_{u_i}, the other one is the remaining users that user u_i has not followed (i.e., the negative user set), denoted as N_{u_i}. Besides, it also has two event sets: one consists of the attended events (i.e., the positive event set), denoted as EP_{u_i}, the other one includes the remaining events (i.e., the negative event set), denoted as EN_{u_i}. Our model assumes that a user prefers the items in the positive set to these in the negative set, which is equal to the following formulas:

$$\begin{aligned} r^u(u_i, u_j) > r^u(u_i, u_k) \ \ &\forall u_i \in U, u_j \in P_{u_i}, u_k \in N_{u_i}, \\ r^e(u_i, e_j) > r^e(u_i, e_k) \ \ &\forall u_i \in U, e_j \in EP_{u_i}, e_k \in EN_{u_i}. \end{aligned}$$

Let $p(r^u(u_i, u_j) > r^u(u_i, u_k))$ and $p(r^e(u_i, e_j) > r^e(u_i, e_k))$ denote the probability that user u_i prefers user u_j to user u_k and the probability that user u_i

prefers event e_j to event u_k. Then, we define $p(r^u(u_i, u_j) > r^u(u_i, u_k))$ and $p(r^e(u_i, e_j) > r^e(u_i, e_k))$ as follows:

$$p(r^u(u_i, u_j) > r^u(u_i, u_k)) := \varepsilon(r^u_{ijk}),$$
$$p(r^e(u_i, e_j) > r^e(u_i, e_k)) := \varepsilon(r^e_{ijk}),$$

where

$$\begin{aligned}
\varepsilon(x) &:= \frac{1}{1+e^{-x}}, \\
r^u_{ijk} &:= r^u(u_i, u_j) - r^u(u_i, u_k) \\
&= w^T(z_{ij} - z_{ik}) + bu_j - bu_k + x_i^T H(x_j - x_k), \\
r^e_{ijk} &:= r^e(u_i, e_j) - r^e(u_i, e_k) \\
&= be_j - be_k + x_i^T(y_j - y_k).
\end{aligned}$$

Let $\Theta = (X, H, Y, bu, be, w)$ denote the parameters of our model, where $X \in \mathbb{R}^{d\times n}$ and $Y \in \mathbb{R}^{d\times m}$ denote the latent user feature matrix and the latent event feature matrix, and $\Phi = (\sigma_x^2, \sigma_h^2, \sigma_w^2, \sigma_{bu}^2, \sigma_{be}^2, \sigma_y^2)$ denote the prior parameters of Θ. By assuming that users are independent to each other and each training sample is also independent, we aim to maximize the following posterior probability:

$$\begin{aligned}
&p(\Theta|R, V, \Phi) \propto \\
&\prod_{u_i\in U, u_j\in P_{u_i}, u_k\in N_{u_i}} p(r^u(u_i, u_j) > r^u(u_i, u_k)|\Theta)\cdot \\
&\prod_{u_i\in U, e_j\in EP_{u_i}, e_k\in EN_{u_i}} p(r^e(u_i, e_j) > r^e(u_i, e_k)|\Theta)\cdot p(\Theta|\Phi),
\end{aligned} \tag{5}$$

where R and V denote the online social network and the offline event participation network, respectively. To avoid over-fitting, we introduce Gaussian priors with zero-mean on the parameters Θ. After applying logarithmic function on Eq. (5), maximizing the posterior probability can be equivalent to minimizing the following objective function:

$$\begin{aligned}
E = &-\sum_{u_i\in U}\sum_{u_j\in P_{u_i}}\sum_{u_k\in N_{u_i}} \ln\varepsilon(r^u_{ijk}) \\
&-\sum_{u_i\in U}\sum_{e_j\in EP_{u_i}}\sum_{e_k\in EN_{u_i}} \ln\varepsilon(r^e_{ijk}) + \frac{\lambda_x}{2}\|X\|_F^2 + \frac{\lambda_y}{2}\|Y\|_F^2 \\
&+ \frac{\lambda_h}{2}\|H\|_F^2 + \frac{\lambda_{bu}}{2}\|bu\|_F^2 + \frac{\lambda_{be}}{2}\|be\|_F^2 + \frac{\lambda_w}{2}\|w\|_F^2,
\end{aligned} \tag{6}$$

where $\lambda_x = 1/\sigma_x^2$, $\lambda_y = 1/\sigma_y^2$, $\lambda_h = 1/\sigma_h^2$, $\lambda_{bu} = 1/\sigma_{bu}^2$, $\lambda_{be} = 1/\sigma_{be}^2$, $\lambda_w = 1/\sigma_w^2$, and $\|\cdot\|_F$ is the Frobenius norm.

3.4 Parameter Learning

To learn the parameters of our model, we use the stochastic gradient descent (SGD) algorithm to minimize the Eq. (6), since it provides fast convergence to the local optimums and has good expandability. In employing SGD, we randomly choose an instance from the training samples. For each selected training instance, we calculate its partial derivative and update the parameters Θ as follows:

$$\Theta \leftarrow \Theta - \alpha * \frac{\partial E}{\partial \Theta},$$

where α is the learning rate. In updating the parameters iteratively, each parameter moves along the descending gradient direction until it converges or the maximum number of iterations is reached.

4 Experiments

4.1 Experimental Setup

Data Allocation. We use the same datasets used in our data analysis for performance evaluation. The details of these two datasets have been shown in Sect. 2.3. For both the datasets, we use s-fold cross validation to evaluate the performance. In detail, we randomly divide each user u_i's positive user set P_{u_i} into s subsets and repeat the holdout method s times. Each time, one of the s subsets is used as the testing set, denoted as $S_{u_i}^{test}$, and the other s-1 subsets form a positive training user set, denoted as $S_{P_{u_i}}^{train}$. We combine $S_{u_i}^{test}$ and its original negative user set N_{u_i} into a new negative training user set, denoted as $S_{N_{u_i}}^{train}$. Our model is trained based on each user's training sets $S_{P_{u_i}}^{train}$ and $S_{N_{u_i}}^{train}$, and the recommendation performance is evaluated on each user's testing set $S_{u_i}^{test}$. Finally, we take the average performance results of all the trials. In this work, we use 5-fold cross validation.

Evaluation Metrics. To evaluate the recommendation performance, we adopt three widely used evaluation metrics: AUC, Precision@k, and MAP@n.

AUC is especially suited for measuring the overall performance in highly imbalanced dataset, as in our case where the number of a user's followees is usually very small with regard to the total number of the users. In our problem, the AUC is defined as follows:

$$AUC = \frac{\sum\limits_{u_i \in U} \sum\limits_{u_j \in S_{u_i}^{test}} \sum\limits_{u_k \in N_{u_i}} \delta(r^u(u_i, u_j) > r^u(u_i, u_k))}{\sum\limits_{u_i \in U} \left|S_{u_i}^{test}\right| \cdot |N_{u_i}|},$$

where $\delta(x)$ is an indicator function which is equal to 1 if x is true and equal to 0 otherwise.

Precision@k and MAP@n are mainly used to evaluate the performance of top-k recommendation. In our problem, the Precision@k measures the ratio of users in the top-k recommendation list that are corresponding to the followees in users' testing sets. MAP@n is the mean of all the users' Average Precision (AP) scores at position n. The AP@n score of each user is defined as follows:

$$AP_{u_i}@n = \frac{\sum_{k=1}^{n} \text{Precision}@k \times I(k)}{|S_{u_i}^{test}|},$$

where $I(k)$ is an indicator function which is equal to 1 if the user at rank k is in user u_i's testing set $S_{u_i}^{test}$ and equal to 0 otherwise. To measure the overall results of recommendation, we set n to the maximum value for MAP@n. For convenience, in the following, we omit the argument n in MAP@n and replace Precision@k with P@k.

Comparison Methods. Since our proposed method, denoted as HNFR, combines all the explicit features (social features and event-based features) and latent features, we devise methods that incorporate these factors individually. Beside, we compare our method with some existing works which are designed for followee recommendation in traditional social networks. In summary, we compare HNFR with the following methods:

- **FoF** [5]**:** For each user u, this method ranks the candidates according to the number of user u's followees who have followed the candidate.
- **CB-MF** [21]**:** This method first employs a LDA-based method on the social relations to discover communities and then applies WRMF [9] on each discovered community.
- **BPR-SF:** This method only uses the social features described in Sect. 3.1 and employs BPR for optimization.
- **BPR-EF:** Similar to BPR-SF, this method only considers the event-based features introduced in Sect. 3.1.
- **BPR-AF:** This method combines all the social features and event-based features, which can be used to verify the effectiveness of integrating all the explicit features.
- **BPR-MF:** This is the basic matrix factorization model for followee recommendation discussed in Sect. 3.2, which only considers the latent features in the online social network and applies BPR for optimization.
- **BPR-MAF:** This method uses the explicit and latent features in the online social network, however, it does not consider the latent features in the offline event participation network.

In our experiment, all the methods are implemented with the LibRec [6] library using JAVA.

Parameter Settings. Empirically, for the regularization parameters λ_x, λ_y, λ_h, λ_{bu}, and λ_{be}, we set all of them to 10^{-2}. Specially, we set the regularization parameter λ_w to 10^{-3}. Besides, we empirically set the dimension of latent features to 20. In both the datasets, we set the initial learning rate α to 10^{-2} for the methods containing the latent features and 10^{-4} for the methods that only use the explicit features. Note that, the LibRec library will adjust the learning rate automatically during the training process.

4.2 Experimental Results

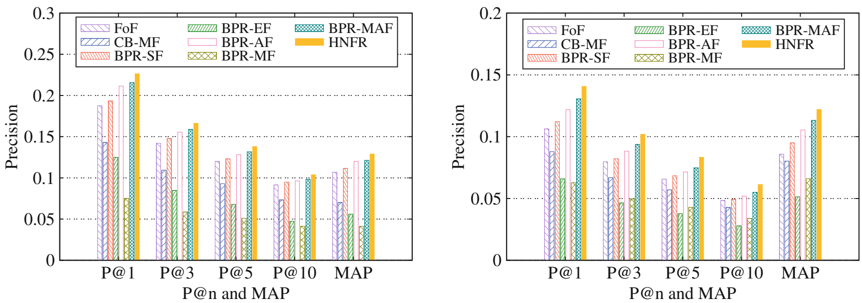
Performance Comparison Under the AUC Metric. In this section, we evaluate the performance of all the methods under the AUC metric. The AUC metric can reflect the overall performance under pairwise ranking.

The AUC results of all the methods in both datasets are shown in Table 2. We can clearly observe that our proposed model, HNFR, always outperforms all the baseline methods in both the datasets significantly. Moreover, the performance of HNFR is better than BPR-MAF, which demonstrates the effectiveness of considering the latent features in the offline event participation network. In both the datasets, BPR-AF achieves better performance than BPR-SF and BPR-EF, which indicates the strength of combining social features and event-based features. We also find that matrix factorization based methods such as BPR-MF and BPR-MAF perform much better than the methods which only consider explicit features like BPR-SF, BPR-EF, and BPR-AF. Besides, we observe that the performance of most methods in Plancast dataset is much better than those in DoubanEvent dataset, which might be due to that the density of the online social network in Plancast dataset is denser than that in DoubanEvent dataset.

Performance Comparison in Top-*k* Recommendation. In this section, we evaluate the overall performance of all the methods in top-k recommendation.

Table 2. The AUC results

	FoF	CB-MF	BPR-SF	BPR-EF	BPR-AF	BPR-MF	BPR-MAF	HNFR
Plancast	0.84236	0.89496	0.84455	0.60117	0.86309	0.92363	0.92959	**0.94738**
DoubanEvent	0.75785	0.78461	0.75888	0.72511	0.80788	0.88330	0.88560	**0.91735**



(a) Plancast 5-fold cross-validation (b) DoubanEvent 5-fold cross-validation

Fig. 3. The P@k and MAP performance

As users mainly focus on the top of the recommendation results, we evaluate the P@k performance under $k = 1, 3, 5, 10$. Meanwhile, we use MAP to measure the overall recommendation ranking list.

The P@k and MAP performance of all the methods in both datasets are shown in Fig. 3. As shown, our method HNFR also achieves the best performance in all the tests, which verifies the effectiveness of our proposed method in followee recommendation in EBSNs. Moreover, we observe that BPR-MF performs much worse than the methods that only consider explicit features, which is contrary to the performance under AUC metric. This may be caused by the following reasons: (1) explicit features play an important role in top-k recommendation; (2) pairwise ranking learning strategy does not focus on the top-k recommendation results. Analogously, we also observe that most methods in Plancast dataset perform better than the corresponding methods in DoubanEvent dataset.

5 Related Work

In this section, we briefly review the related works, including followee/friend recommendation and collaborative filtering.

Followee/Friend Recommendation. The problem of followee/friend recommendation in social networks, which can also be regarded as a special kind of link prediction problem that foucuses on predicting the social links among users, has been widely studies for many years [2,5,7,8,12,14,18–21]. Liben-Nowell et al. [12] first study the link prediction problem in social networks. They develop and compare several unsupervised methods based node proximity in a social network, such as methods based on node neighborhoods and methods based on the ensemble of all paths. Zhao et al. [21] propose a community-based followee recommendation in Twitter-style social networks. They first employ an LDA-based community discovery method based on the followers/folloewee relations and then apply matrix factorization method on the discovered communities to recommend followees.

In addition to using the network structure information, there are many works exploiting other types of information such as user-generated content, user profile, and locations. For example, Chen et al. [5] study the people recommendation problem in an enterprise social networking and propose four recommendation methods based on social network structure or user-generated contents. They find that methods based on social network structure are good at recommending known contacts for users while methods based on user-generated content are stronger in discovering new friends. Moreover, Hannon et al. [8] propose to recommend followees in Twitter based on user profiling. They evaluate different profiling strategies based on the tweets or relations of users' Twitter social graphs and find that the collaborative strategies perform better than the content strategies. In addition, Yuan et al. [20] study how to exploit sentiment homophily for link prediction. They propose a topic-sentiment affiliation based graphical model which incorporates the sentiment features extracted from tweets, structural features based on social graph, and topical features based on the topical affiliation of two users.

Recently, there exist some studies focusing on different friend recommendation problem to satisfy different needs. For instance, Barbieri et al. [2] study the problem of link prediction with explanations for user recommendation. They propose a stochastic topic model over directed and nodes-attributed graphs which can produce different types of explanation for different kinds of links (a topical link or a social link). In addition, Wan et al. [19] study the problem of informational friend recommendation which aims to recommend friends according to users' informational needs. They first employ collaborative filtering method to predict a user's rating for each post and then rank the candidate users based on their informational utilities.

In this work, we propose to exploit the offline event participation information for followee recommendation, which is a unique and important characteristic of EBSNs, and no previous works in followee recommendation have considered such informantion. Moreover, we design a novel recommendation model utilizing all the latent and explicit features of both the social relations and the offline event participation records, which is also different from previous works.

Collaborative Filtering. Collaborative filtering is a main recommendation method, which has been widely employed in recommendation systems. [3,4,9–11,15–17]. Matrix factorization plays an important role in collaborative filtering techniques. The basic concept of matrix factorization is to seek latent representations for both items and users, which are usually low-dimensional vectors of factors in the latent space. In these works, Koren et al. [10] propose to incorporate bias factors, temporal dynamics, or confidence levels into matrix factorization. Salakhutdinov et al. [17] propose probabilistic algorithms for matrix factorization which scale linearly with the number of observations. Lee et al. [11] propose a matrix factorization algorithm with non-negativity constraints which produces a parts-based representation of the original matrix.

There are some works focusing on the matrix factorization based recommendation with implicit feadbacks [9,15,16]. For example, Hu et al. [9] proposed a factor model which treats the implicit feedbacks as indication of positive and negative preference associated with vastly varying confidence levels. Rendle et al. [16] propose a Bayesian optimization criterion named BPR for personal ranking from implicit feedbacks and apply it to matrix factorization. Unlike previous works, they adopted pairwise user preference towards items, which have the underlying assumption that a user prefers item viewed by the user to all other non-observed items.

In this work, inspired by BPR, we propose a new Bayesian optimization framework which adopts pairwise user preference on both the online social network and the offline event participation network.

6 Conclusion

In this paper, we study the problem of followee recommendation in EBSNs. We propose a new followee recommendation method called HNFR, which exploits the heterogeneous nature of EBSNs. In our method, we combine all the explicit and latent features which are captured from the online social network and the offline event participation network. Moreover, we propose a Bayesian optimization framework which adopts pairwise user preference on both the social relations and the events. The experimental results on real-world data demonstrate the effectiveness of our method.

Acknowledgements. The work was supported by National Natural Science Foundation of China under Grant 61502047.

References

1. Adamic, L.A., Adar, E.: Friends and neighbors on the web. Soc. Netw. **25**(3), 211–230 (2003)
2. Barbieri, N., Bonchi, F., Manco, G.: Who to follow and why: link prediction with explanations. In: SIGKDD, pp. 1266–1275 (2014)
3. Cai, Y., Lau, R.Y.K., Liao, S.S.Y., Li, C., Leung, H., Ma, L.C.K.: Object typicality for effective web of things recommendations. Decis. Support Syst. **63**, 52–63 (2014)
4. Cai, Y., Leung, H., Li, Q., Min, H., Tang, J., Li, J.: Typicality-based collaborative filtering recommendation. IEEE Trans. Knowl. Data Eng. **26**(3), 766–779 (2014)
5. Chen, J., Geyer, W., Dugan, C., Muller, M.J., Guy, I.: Make new friends, but keep the old: recommending people on social networking sites. In: CHI, pp. 201–210 (2009)
6. Guo, G., Zhang, J., Sun, Z., Yorke-Smith, N.: Librec: a java library for recommender systems. In: Posters, Demos, Late-breaking Results and Workshop Proceedings of User Modeling, Adaptation, and Personalization (UMAP 2015) (2015)
7. Guy, I., Ronen, I., Wilcox, E.: Do you know?: recommending people to invite into your social network. In: IUI, pp. 77–86 (2009)
8. Hannon, J., Bennett, M., Smyth, B.: Recommending twitter users to follow using content and collaborative filtering approaches. In: RecSys, pp. 199–206 (2010)
9. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: ICDM, pp. 263–272 (2008)
10. Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Comput. **42**(8), 30–37 (2009)
11. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: NIPS, pp. 556–562 (2000)
12. Liben-Nowell, D., Kleinberg, J.M.: The link prediction problem for social networks. In: CIKM, pp. 556–559 (2003)
13. Liu, X., He, Q., Tian, Y., Lee, W., McPherson, J., Han, J.: Event-based social networks: linking the online and offline socialworlds. In: SIGKDD, pp. 1032–1040 (2012)
14. Menon, A.K., Elkan, C.: Link prediction via matrix factorization. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) ECML PKDD 2011, Part II. LNCS, vol. 6912, pp. 437–452. Springer, Heidelberg (2011)

15. Pan, R., Zhou, Y., Cao, B., Liu, N.N., Lukose, R.M., Scholz, M., Yang, Q.: One-class collaborative filtering. In: ICDM, pp. 502–511 (2008)
16. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: bayesian personalized ranking from implicit feedback. In: UAI, pp. 452–461 (2009)
17. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: NIPS, pp. 1257–1264 (2007)
18. Scellato, S., Noulas, A., Mascolo, C.: Exploiting place features in link prediction on location-based social networks. In: SIGKDD, pp. 1046–1054 (2011)
19. Wan, S., Lan, Y., Guo, J., Fan, C., Cheng, X.: Informational friend recommendation in social media. In: SIGIR, pp. 1045–1048 (2013)
20. Yuan, G., Murukannaiah, P.K., Zhang, Z., Singh, M.P.: Exploiting sentiment homophily for link prediction. In: RecSys, pp. 17–24 (2014)
21. Zhao, G., Lee, M., Hsu, W., Chen, W., Hu, H.: Community-based user recommendation in uni-directional social networks. In: CIKM, pp. 189–198 (2013)