# Multi-feature based event recommendation in Event-Based Social Network *

**Jiuxin Cao**[†]**, Ziqing Zhu , Liang Shi , Bo Liu , Zhuo Ma**

*School of Computer Science and Engineering, Southeast University,*
*Nanjing, 211189, Jiangsu, China*

*Key Laboratory of Computer Network and Information Integration, Ministry of Education,*
*Nanjing, 211189, Jiangsu, China*

*E-mail: jx.cao@seu.edu.cn, zzqxztc@seu.edu.cn, shiliangdn@seu.edu.cn, bliu@seu.edu.cn,*
*mazhuo@seu.edu.cn*

**Abstract**

As a new type of heterogeneous social network, Event-Based Social Network (EBSN) has experienced rapid development after its appearance. In EBSN, the interaction data between users and events is relatively sparse because of the short life cycle of events, which brings great challenges to event recommendation. In this paper, a multiple features based event recommendation method is proposed, which makes full use of various information in the network to mine users' preference for event recommendation. Firstly, a heterogeneous information network model is constructed based on the intrinsic structure characteristics. Then multiple features about topology, temporal, spatial and semantic are extracted to measure the user's event preference, and a linear scoring model is designed to acquire user's preference score on events. At last, the bayesian personalized ranking method is used to learn the feature weights by using user-event pairs in scoring model and events are recommended to users according to the descending score order. Experiments are carried out on two real EBSN data sets, the results show that our approach can effectively alleviate the data sparseness problem and achieve better recommendation results.

*Keywords:* Event-Based Social Network, feature analysis, scoring model , event recommendation

## 1. Introduction

As traditional social networks have experienced rapid development in recent years, some special social networks have been appearing, such as Location-based Social Network (LBSN)[1] which is based on location with check in function and Picture-based social networks (PBSN)[2] with Flickr as a representative. EBSN[3] is another type of complex and heterogeneous social network, which aims to create connections between strangers rather than acquaintances using a way of user-self-organization with Meetup as the typical representative. EBSN is not only an online sharing platform providing services for users, but also helping users organize offline events, which brings convenience to users to carry out face to face communication.

EBSN has experienced rapid development after

---

* First and second authors contributed equally to this paper
[†] Corresponding author

its appearance. Taken Meetup as an example, the platform has attracted more than million users so far and has more than 200 thousand online groups. Thousands of events are published on the platform every day. Facing so many events, it will be time consuming and may not be able to achieve very good results if users search for their interesting events by themselves. In such case, we need to use personalized recommendation technology to recommend interested events to users which can help solve the event selection problem and improve the quality of platform service.

Personalized recommendation is a hot issue in social network and related works [4–13] show that adding social relation and attribute information can effectively improve the performance of recommendation system, such as topology infomation[7,8], community structure[9–11], semantic content[12,13] and so on. The structure of social relations in EBSN is different from the two typical social relationship structure in traditional social networks: the social graph structure represented by Facebook and the interest graph structure represented by Twitter. In EBSN, the connections between users are generated through participating in a group or an event. Belonging to the same group or event represents high similarity between users' interest in a certain extent. So when carrying out personalized event recommendation in EBSN, we should consider a proper way to fuse the online social relation of group and the offline social relation of event to help improve the recommendation effect. Liu et al.[3] proposed the definition of EBSN and at the same time they carried out data analysis on the network which shows that the behavior of users' participation in events has certain rules to follow in time and space. Whereas these temporal and spatial information has received little attention for recommendation in EBSN, these can help reflect users' willingness to participate in an event.

In this paper, we analyze topological structure, spatio-temporal information, and semantic content to obtain various features. Meanwhile, according to our knowledge, most researches combined some of them in some way, few works took an insight look into these together, therefor we propose a multiple hybrid features based event recommendation

method, which makes use of various features. What is more, the data imbalance problem[37,38] also exist in EBSN, that users only choose a few events they like in social network(few positive simples). Although users only choose some groups or events, it does not mean users do not like the rest events, we take advantage of bayesian personalized ranking (BPR) learning algorithm[36] to use all sample data more comprehensively to solve this problem. Our approach measures user's willingness of participating in an event from multiple perspectives. It will recommend events to users according to the events' scores. The approach can alleviate the data sparsity of EBSN by using multi-dimensional information, and improve the final performance of the recommendation. Our main contributions of this paper are as follows:

(1)Meta-path construction method is applied to get topological feature based on association relations between users and events, which can be used to measure the weight of topological relations.

(2)To mine the users' behavior regularities, time frames mechanism is defined to show temporal feature of users' events attending .

(3)Due to the impact of spatial distance on users' movement, a heavy tailed power-law distribution is fitted based on the distance from users' location to events' location, to explore the patterns of users' movement. Thus, we obtain the spatial feature of users.

(4)Semantic association between user and event is evaluated by the similarity between semantic vectors which are derived from latent semantic analysis (LSA) of semantic content.

(5)A multi-feature based scoring model is put forward, whose parameters are learnt by BPR algorithm.

In the remainder of this paper, Section 2 reviews related works of event recommendation. Section 3 gives description of the experiment data set and constructs the network model of EBSN. Section 4 describes our proposed scoring model based on multiple features and the event recommendation algorithm based on the scoring model. The method to learn the feature weights of scoring model is presented in Section 5, followed by experimental eval-

uation in Section 6. At last, conclusions and future works are provided in Section 7.

## 2. Related Works

EBSN has received more and more attention and related works is gradually increasing in recent years, the works are mainly concentrated in the following four aspects: recommendation[15–23], influence analysis[24], user behavior prediction[25,26] and event organization[27,28]. Among them, related works of recommendation is the most popular, mainly including event recommendation, group recommendation and tag recommendation.

Related works about event recommendation had already been conducted before the appearance of EBSN. Considering the trust plays important role when generating event recommendations in Social Networks, Lee et al.[29] proposed a collaborative filtering method based on the trust information between users for culture event recommendation. He used the PITTCULT website as the research platform to get the trust information. The results showed that the trust information can help promote the effect of recommendation. And De Meo et al.[14] designed a model to represent and handled trust and reputation in a social internetworking system, so that they proposed an approach that exploits these parameters to compute the reliability of a user or a social network, as well as the quality of a resource(event). Finally, they can recommend quality event to user. In addition, Minkov et al.[30] proposed a collaborative ordering method for event recommendation by mixing user's own historical preference and other user's preference. Liao et al.[31] constructed three social relation networks using attribute and structure information in the network, using the factor graph model to combine the three relation networks together for event recommendation. The above research on the event recommendation is not for the EBSN network environment, and dose not consider the characteristics of EBSN, such as online and offline social relationships, temporal and spatial attribute information, and so on.

In recent years, research on event recommendation for EBSN network environment increased year by year. According to the purpose of recommendation, these works can be divided into three categories:

(1) User as the center. Recommending interested events to users according to their preference.

(2) Event as the center. Recommending users who prefer to participate in events.

(3) Group as the center. Mining the group preference and recommending events to the group users.

Our research goal is more close to the first two categories above.

According to the recommendation method, the research work can be summarized as three aspects:

(1) Recommendation based on latent factor mining. This kind of approaches first carry out matrix factorization based on the interaction information in the network. Then the implicit feature vectors of users and events are trained and extracted. At last, recommendation is done according to the scores based on feature vectors. Qiao et al.[16] proposed the user scoring model for events by matrix factorization model with spatial attribute. They obtained the implicit feature vectors of users and events through data training, and recommended users for event based its predicted score which makes use of vectors and the spatial attribute information. Zhang et al.[18] proposed a collective bayesian poisson factorization model for event recommendation using different kinds of relation information in the network, such as the social relation, the relation between users and events, and the relation between every event textual contents.

(2) Recommendation based on context analysis. This kind of methods make use of the attributes of nodes and the direct relation information in the network to construct recommendation model. Wang et al.[19] use content feature, social relation feature and local popularity feature to construct their event recommendation model and recommend events to users according to the score order based on these features. Augusto et al.[20] constructed the contextual model by combining the social feature obtained by multiple social relation factorization and other attribute information, then used a linear model to conduct the event ranking and recommendation.

(3) Recommendation based on graph model. It

is characterized by making full use of the links between nodes in the network to construct the recommendation model, and finally recommendation is conducted through predicting the probability of the links. Nguyen Pham et al.[21] used multivariate Markov chain model to model the links on a variety of attributes between various nodes in the network. They calculated the probability of the unknown links based on the transition probability of various links and conducted recommendation according to the predicted probability.

From the existing research works, we can find that many researchers focus on abstracting features and building model to utilize various of information more effectively, due to data sparsity in EBSN. Researchers have proposed several methods to solve different application problem based on alleviating data sparsity, such as making use of collective factorization on multiple relations, analyzing context information in the network and so on. It is not difficulty to understand the interaction data sparsity between users and events because of shorter event life cycle, which has brought great challenge to event recommendation in EBSN. In this paper, we propose a multiple features based event recommendation method which makes use of various information in the network to mine users' preference and alleviate the data sparsity in event recommendation.

## 3. Data Description and Modeling EBSN

### 3.1. Data Description

The data for experiment used in this paper are extracted from the Meetup dataset which was crawled in work 3. Considering the size of the global dataset is too large and regional characteristics of events in EBSN, we extract the data of two popular cites (New York abbreviated as NY and San Francisco abbreviated as SF) from the global dataset to carry out our analysis and experiment. We also filter out those zombie users without event participating record for the data validity. At the same time, we supplement related description information which can not be found in the original dataset about users and events using the public API provided by Meetup website.

The statistics of the two city datasets are shown in Table 1, and the network interaction density measure the density of interaction relation between users and events compared with all possible interaction relation in EBSN.

The basic dataset statistics analysis is carried out on dataset. Fig.1(a) shows the frequency distribution on the number of events per user participates in and Fig.1(b) shows the frequency distribution on the number of groups per user participates in. The frequency distribution on the number of participants per event and the number of members per group shown in Fig.1(c) and Fig.1(d) respectively. All the figures use the double-logarithmic coordinate system with base 10. From the figures, we can see that the degree distributions of related links of various nodes in the network are in line with power-law distribution and have a typical heavy-tailed characteristics which are consistent with the conclusion of existing related work on complex social networks.
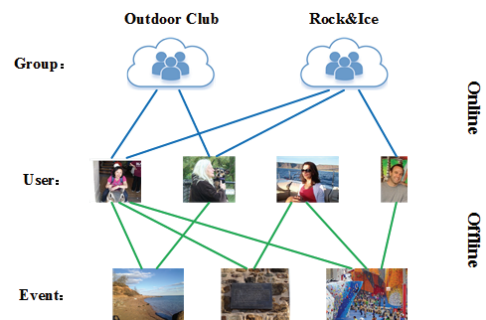
### 3.2. Modeling EBSN



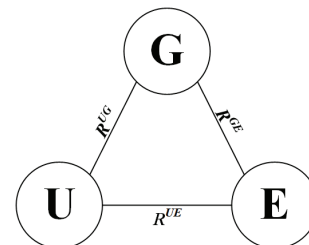Fig. 2. Interaction Relation in EBSN



Fig. 3. Network Schema of EBSN

Through analysis on the EBSN dataset, there are only relation links based on interaction records between nodes without direction and weight in the EBSN dataset, shown in Fig.2. We model
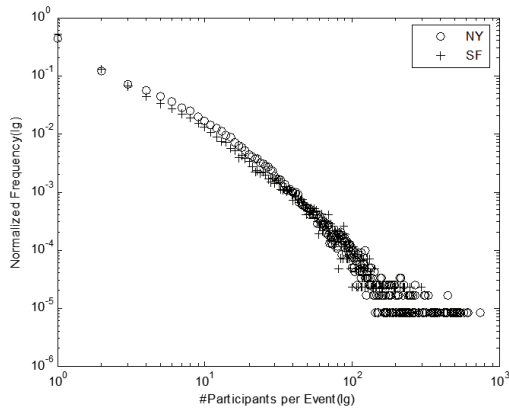
Table 1. DATASET STATISTICS

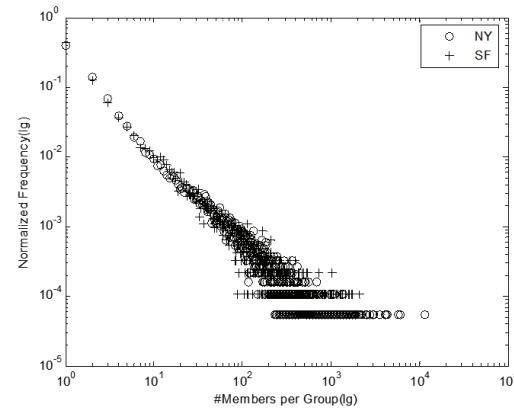| Dataset | NY | SF |
|---|---|---|
| #Users | 129407 | 37963 |
| #Events | 121111 | 42945 |
| #Groups | 19636 | 10441 |
| #Records between users and events | 782484 | 218115 |
| #Records between users and groups | 744517 | 228491 |
| #Network interaction density | $4.99 \times 10^{-5}$ | $1.4 \times 10^{-4}$ |
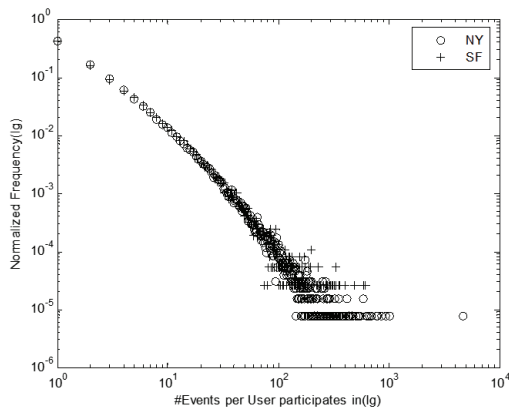


(a) #Events per User
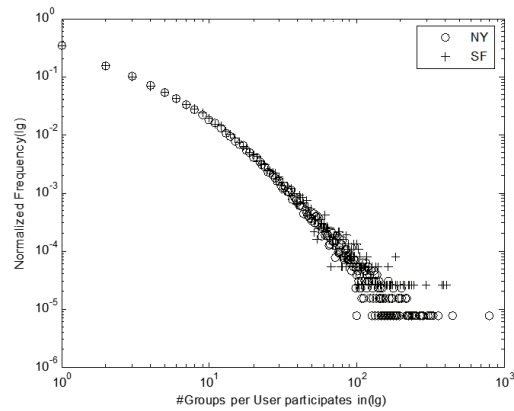
(b) #Groups per User

(c) #Participants per Event

(d) #Members per Group

Figure 1: Statistics Analysis of Frequency Distribution on Dataset

the EBSN as a heterogeneous information network without direction and weight, represented as $N = (V, L, A, T, R)$, where $V$ represents the node set. $L$ represents the edge set. $A$ represents the attribute set, including the attributes of various nodes, such as the location attribute of user's home, the temporal attribute of event, the location attribute of event and so on. $T$ is the set of types of nodes, mainly including three types: User ($U$), Event ($E$) and Group ($G$), and $R$ is the set of types of edges, mainly in-

cluding three types of indirection edges: User-Event Edge ($R^{UE}$), User-Group Edge ($R^{UG}$) and Group-Event Edge ($R^{GE}$), shown in Fig.3.

## 4. Multiple Hybrids Features based Event Recommendation Method

### 4.1. Modeling Multiple Features

In order to alleviate the impact of EBSN data sparsity, we make full use of the online and offline social relations and various node information to extract four types of features for further constructing recommendation scoring model. The four features include : (a) topological feature, (b) temporal feature, (c) spatial feature and (d) semantic feature. The following gives the description of four features' modeling.

#### 4.1.1. Modeling Topological Feature

Compared to other information, the network topology information is accessible and objective. As the relations in EBSN dataset are relatively sparse, it's necessary to mine information behind the indirect relations which can characterize the user's preference. Considering the large size of the network, traditional methods such as random walk on the entire network are not practical. In order to mine user's preference on events by different types of paths in the network, we introduce the concept of **meta-path**[32] which can reflect user's preference. Meta-path is defined as below:

**Definition 1.** Meta-path. For network schema $N_s = (T, R)$, a meta-path is defined as the path schema like $P = T_0 \xrightarrow{R_0} T_1 \xrightarrow{R_1} \cdots \xrightarrow{R_{n-1}} T_n$, where $T_i \in \{U, G, E\}$, $R_i \in \{R^{UE}, R^{UG}, R^{GE}\}$. In fact, Meta-path is an abstract of a type of paths in the network, and its length of a meta-path equals the number of edges in it. Through meta-path, the relationship between network nodes under different kinds of paths with different length can be obtained. And much more meta-paths is by inserting path fragment U-E-U and U-G-U into original meta-path to get more indirect relation information in EBSN.

For two nodes of any type, there maybe meta-paths with various types and lengths, and the correlation will decrease with the path length increas-

ing. Huang et al.[33] also used meta-path feature to do link prediction and the experiment result showed that meta-paths with length less than five are enough for the effect. Considering the balance of the effect and the time complexity, nine meta-paths are selected as the topology feature set, shown as Table 2.

Before using meta-paths to measure correlation between nodes, the meta-path features should be quantified. We introduce the definition of instance-path firstly and then give the computing method of the meta-path eigenvalue based on instance-path.

**Definition 2.** Instance-path. For network entity $N_G = \{V, L, A\}$ and a meta-path $P = T_0 \xrightarrow{R_0} T_1 \xrightarrow{R_1} \cdots \xrightarrow{R_{n-1}} T_n$, if a real path $p = (v_0, v_1, v_2, \cdots, v_n)$ meets the condition: $v_i \in V$, $f_\phi(v_i) = T_i$ and $f_\psi(v_i, v_{i+1}) = R_i$, where $f_\phi$ and $f_\psi$ are type mapping function of nodes and edges. $p$ is an instance-path of meta-path $P$.

In the application scenarios of this paper, the meta-paths start from $U$ and end with $E$. For a meta-path $P = T_0 \xrightarrow{R_0} T_1 \xrightarrow{R_1} \cdots \xrightarrow{R_{n-1}} T_n$, we can find the instance-paths set $P'$, where the path in $P'$ starts from $v_0$ and ends with $v_n$ and meets the condition: $v_0 \in U$ and $f_\phi(v_0) = T_0$, $v_n \in E$ and $f_\phi(v_n) = T_n$. According to $P'$, the eigenvalue of meta-path $P$ between $v_0$ and $v_n$ is defined, which equals the sum of eigenvalues of instance-paths in $P'$, denoted as $Rv(P)$:

$$Rv(P) = \sum_{p \in P'} Irv(p) \tag{1}$$

where $Irv(p)$ stands for the eigenvalue of instance-path $p$.

We use the method based on random walk to compute the eigenvalue of instance-path. For an instance-path $p = (v_0, v_1, \cdots, v_n)$ , we define $Irv(p)$ as a walking from $v_0$ to $v_n$ along the path $p$ randomly. According to the assumption that steps in random walk process are independent, $Irv(p)$ is defined as below:

$$Irv(p) = \prod_{i=0}^{n-1} Pro(v_i, v_{i+1}) \tag{2}$$

where $Pro(v_i, v_{i+1})$ is the probability of the step from $v_i$ to $v_{i+1}$ in the random walk process. Considering the edges without weight in EBSN,

Table 2. NINE META PATHS IN THE NETWORK

|   | Meta-path | Path Semantic (User may be interested in Event) |
|---|---|---|
| 1 | U—G—E | events in user's groups |
| 2 | U—G—U—E | events that other users of the same group have participated in |
| 3 | U—E—U—E | events that other users of the same group have participated in |
| 4 | U—G—U—G—U—E | Similar to path 2, with double U—G—U |
| 5 | U—G—U—G—E | Fusing path 2 and path 1 to get the semantic |
| 6 | U—G—U—E—U—E | Fusing path 2 and path 3 to get the semantic |
| 7 | U—E—U—G—U—E | Fusing path 3 and path 2 to get the semantic |
| 8 | U—E—U—G—E | Fusing path 3 and path 1 to get the semantic |
| 9 | U—E—U—E—U—E | Similar to path 3, with double U—E—U |

$Pro(v_i, v_{i+1})$ is defined as below:

$$Pro(v_i, v_{i+1}) = \frac{1}{|N(v_i)|} \qquad (3)$$

where $N(v_i)$ denotes the set of the neighbor nodes of $v_i$ with the same type as $v_{i+1}$. Considering the above definition may bring the problem of relation strength decreasing as the size of group or event increasing, we give another definition for $Pro(v_i, v_{i+1})$:

$$Pro(v_i, v_{i+1}) = \frac{1}{1 + \ln(|N(v_i)|)} \qquad (4)$$

The effects of the above two computing methods will be compared through experiments, and the meta-path based topology feature set is denoted as $\psi^P$.

### 4.1.2. Modeling Temporal Feature

User's behavior often has certain regularities in time dimension, for example, some users often go out to participate in social events in the evening of weekend, and others like relaxing on weekday. We use the users' events records to find the temporal regularity of behavior and assume that users will be in line with their temporal regularity in the future. Considering the data sparsity, enhancing the granularity of time analysis is needed.
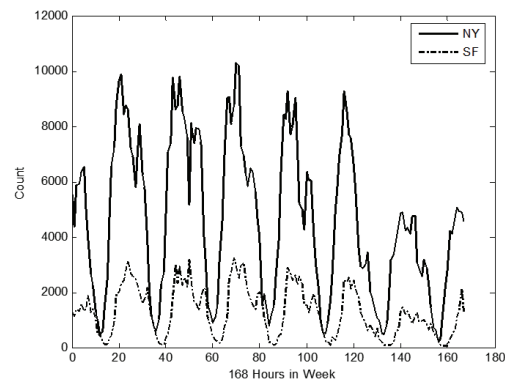


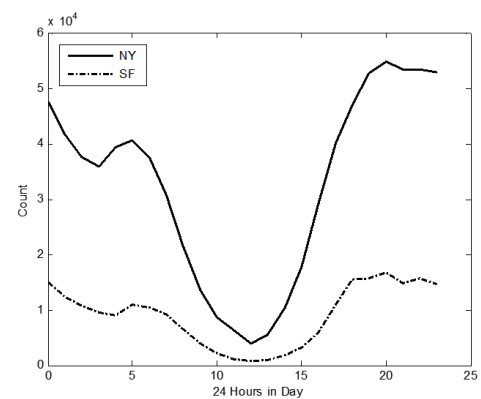Fig. 4. Distribution of Events' Time in Week mode



Fig. 5. Distribution of Events' Time in Day mode

Fig.4 and 5 show the statistics of the time users attended in events in time frame of week (168 hours) mode and day (24 hours) mode on Meetup.com dataset. It can be seen form Fig.4 that user's habit of attending events is different from weekday and weekend, and it is also quite different in different

Table 3: TIME FRAMES IN EBSN

| Time frame | Symbol | Description |
|---|---|---|
| 24 hours in day | $T_1$ | divide day into twenty four hours |
| 7 days in week | $T_2$ | divide week into seven days |
| 3 periods in day | $T_3$ | divide day into three periods: 00:00-7:59, 8:00-15:59 and 16:00-23:59 |
| 6 hybrid time frames of week and day | $T_4$ | divide week time into weekday and weekend |
| | | divide weekday and weekend into three periods |

periods of day in Fig.5. Correspondingly, four time frames are created to mine user's behavior habit, shown as Table 3.

According to the users' history records and the time frames, the eigenvalue of temporal feature is defined under different time frames, shown as Eq.(5):

$$\varphi^{T_i}(u,e) = \frac{|\{e'|e' \in E(u) \wedge f_{T_i}(t(e')) = f_{T_i}(t(e))\}|}{|E(u)|} \tag{5}$$

where $e$ is event to be scored, $e'$ represents event user $u$ has attended, $E(u)$ represents the event set user $u$ has attended, $t(e)$ represents the time of event $e$, $f_{T_i}(t)$ is the function that maps event time to time frame. The features under the four time frames constitute the temporal feature set, denoted as $\psi^T$.

### 4.1.3. Modeling Spatial Feature

Related research[34] shows that user's behavior is often limited by spatial distance. Considering the impact of spatial distance, we make the statistics for the frequency distribution on the distance between events to users' home location, shown as Fig.6.

We model and fit the frequency distribution on the distance. According to the fitting result and the distance between user's home and event's location, the spatial feature is quantified as a probability to measure user's willingness to participate in event. Considering the data sparsity, we assume that the behavior patterns of all users are consistent and do analysis and fitting on the records of all users.

It can be found from Fig.6 that the distribution is divided into two parts by $k$ km(here is 30km), one part of the data is in line with exponential distribution, and another part of the data is consistent

with heavy tailed power-law distribution. So we use the sectional fitting method to fit the distance frequency distribution. The fitting results are denoted as $f_L(f_{L_{NY}}(d)$ and $f_{L_{SF}}(d)$ represents NewYork or San Francisco), shown as Eq.(6) and Eq.(7):

$$f_{L_{NY}}(d) = \begin{cases} 0.1495 * e^{-0.1718*d}, d <= 30km \\ 0.04313 * d^{-1.192}, d > 30km \end{cases} \tag{6}$$

$$f_{L_{SF}}(d) = \begin{cases} 0.1575 * e^{-0.2104*d}, d <= 30km \\ 0.04313 * d^{-1.147}, d > 30km \end{cases} \tag{7}$$

where $d$ represents location distance between user home and event.

The eigenvalue of the spatial feature is defined based on the Eq.(6) or Eq.(7):

$$\varphi^L(u,e) = f_L(dis(u,e)) \tag{8}$$

where $dis(u,e)$ represents the distance between user $u$ and event $e$. This feature constitutes the spatial feature set, denoted as $\psi^L$.
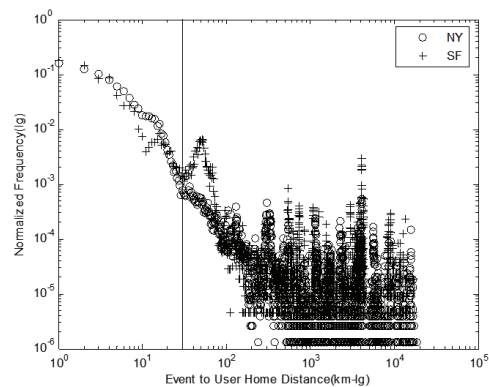


Fig. 6. Frequency Distribution on Distance between Event location and User's Home location

#### 4.1.4. Modeling Semantic Feature

Combining all users' tags, all events' name and description together to construct the text content set, we use Latent Semantic Analysis (LSA) model to mine the latent semantic topic vector of users and events, considering the high dimensional and data sparse attribute of this set. And then the semantic similarity can be measured by computing the cosine similarity between user and event shown as Eq.9.

$$\varphi^S(u,e) = \cos(\theta_u, \theta_e) = \frac{\theta_u \cdot \theta_e}{|\theta_u||\theta_e|} \quad (9)$$

where $\theta_u$ is semantic vector of user $u$ and $\theta_e$ stands for semantic vector of event $e$.

Here the topic number of semantic vector should be first decided owing to it has impact on similarity measurement. Because the core technique of semantic space dimension reduction is Singular Value Decomposition (SVD) in LSA, we analyze the distribution of singular value to get the topic number, ie the latent semantic dimension, shown in Fig. 7.
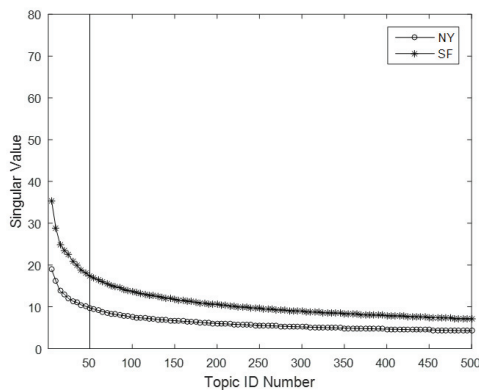


Fig. 7. Distribution of First Five Hundred Singular Value

It can be seen that the singular value drops smoothly about fiftieth topic, considering the balance of corpus size, semantic loss and computing complexity, we choose the topic number 50 for LSA model.

What's more, related research[35] shows that the smaller the average similarity between topics is, the better the effect of the topic model will be. We design the method based on this idea to verify rationality of topic number 50. The specific method is shown as below:

(i) Assuming the extracted words distribution vector of topic $i$ and $j$ is $z_i$ and $z_j$ respectively, we firstly compute the cosine similarity between two topic vectors:

$$sim(z_i, z_j) = \cos(z_i, z_j) = \frac{z_i \cdot z_j}{|z_i||z_j|} \quad (10)$$

(ii) Assuming the topic number is $K$, the average similarity between topics is defined as Eq.(11):

$$avgsim(K) = \frac{2 \times \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} sim(z_i, z_j)}{K \times (K-1)} \quad (11)$$

According to the experiment result, the topic number we selected should have low average similarity, shown in Fig.8. This feature on textual content of users and events constitute the semantic feature set, denoted as $\psi^S$.
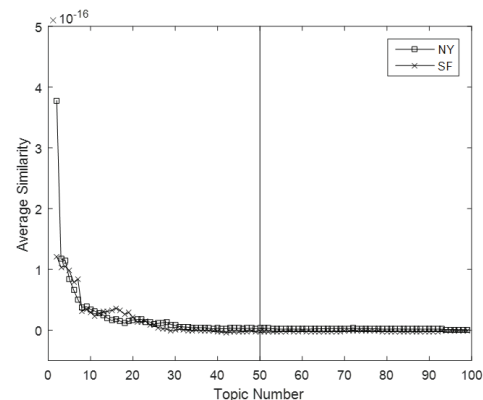


Fig. 8. Experiment on Topic Number

#### 4.2. Scoring Model

According to the extracted features, the recommendation features set is generated by combining them together, denoted as $\psi = \psi^P \cup \psi^T \cup \psi^L \cup \psi^S = \{\psi_1, \psi_2, \dots, \psi_{15}\}$, including nine meta-path features, four temporal features, one spatial feature and one semantic feature. We put all the features into a linear model to construct the scoring model for measuring user's preference or willingness on event. Assuming that for user $u$ and event $e$, the feature vector is $\varphi(u,e) = \{\varphi_1, \varphi_2, \dots, \varphi_{15}\}$, the score of user $u$ to

event $e$ is defined as Eq.(12):

$$r(u,e) = \theta \cdot \varphi = \sum_{k=1}^{15} \theta_k \cdot \varphi_k \qquad (12)$$

where $\theta_k$ represents the weight of the k-th feature. For the weight vector $\theta = \{\theta_1, \theta_2, ..., \theta_{15}\}$ in the scoring model, it needs to be learned through data training. The learning method will be introduced in the following section and recommendation will be carried out on the rank of events' score.

---

**Algorithm 1** Mutil-feature based event recommendation algorithm

---

**Input:** Feature Set $\psi = \psi^P \cup \psi^T \cup \psi^L \cup \psi^S$, EBSN network $G$, Target User $u_0$, Weight Vector $\theta$
**Output:** Event Recommendation List $E'$
  $E = initializeCandidateEventSet(u_0)$
  **for** each $e_i \in E$ **do**
    Array $\varphi$
    **for** each $\psi_j \in \psi$ **do**
      **if** $\psi_j \in \psi^P$ **then**
        $P'_j = findInstancePathSet(u_0, \psi_j)$
        $v_P(u_0, e_i) = 0$
        **for** each instance-path $p_{jk} \in P'_j$ **do**
          $v_P(u_0, e_i) += computeInstancePathValue(p_{jk})$
        **end for**
        $\varphi[j] = v_P(u_0, e_i)$
      **else if** $\psi_j \in \psi^T$ **then**
        $\varphi[j] = computeTimeValue(u_0, e_i, \psi)$
      **else if** $\psi_j \in \psi^L$ **then**
        $\varphi[j] = computeSpatialValue(u_0, e_i, \psi)$
      **else if** $\psi_j \in \psi^S$ **then**
        $\varphi[j] = computeSemanticValue(u_0, e_i, \psi)$
      **end if**
    **end for**
    $r(u_0, e_i) = \sum_{k=1}^{15} \theta_k \cdot \varphi[k]$;
  **end for**
  $E' = Top\text{-}N(E, r)$
  **return** $E'$

---

### 4.3. Recommendation Algorithm

According to the extracted features and the scoring model, recommendation steps are given as below: (i) Determining the candidate event set and the weight vector $\theta$. (ii) Calculating feature vector for each event in the candidate set, as $\varphi$. (iii) Calculating event score $r$ according to the weight vector $\theta$ and the feature vector $\varphi$. (iv) Generating recommenda-

tion list based on the rank of event scores. The specific algorithm is shown as algorithm 1.

## 5. Model Learning

### 5.1. Optimization Objective

In the event scoring model, the feature's weight represents the importance of the feature. We use the implicit feedback as the training data to learn the weight vector. Implicit feedback which only represents the interaction between user and event is different from explicit feedback which represents the weight of relation, such as user's item scoring. If there is interaction between user and event, the feedback is 1, otherwise, the feedback is 0.

Traditional parameter learning methods based on classifier or loss function optimizing can not work well for implicit feedback data. Inspired by Rendle et al.[36], a method based on maximum likelihood estimation called BPR is used to learn the weight vector. The goal of the learning process is to make the order of all user-event pairs correct, which means the events that user has attended should be ranked before other events. According to this idea, we define the Bayesian formulation of the optimization criterion, which is to maximize the posterior probability as below:

$$p(\theta|R) \propto p(R|\theta)p(\theta) \qquad (13)$$

where $\theta$ is the weight vector, $R$ represents the set of all user-event pairs.

For calculability, we assume that users attend events independently from each other and the attending behavior for different events of a user is also independent. According to the assumption, $p(R|\theta)$ can be rewritten as below:

$$\begin{aligned} p(R|\theta) &= \prod_{u \in U} p(R_u|\theta) \\ &= \prod_{u \in U} \prod_{R_u} p(e_i >_u e_j|\theta) \end{aligned} \qquad (14)$$

where $R_u$ represents the set of user-event pairs of user $u$, $p(e_i >_u e_j|\theta)$ represents the probability of event $e_i$ ranked before event $e_j$ for user $u$, which is defined as below:

$$p(e_i >_u e_j|\theta) = \sigma(r(u, e_i) - r(u, e_j)) \qquad (15)$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$.

In order to reduce the number of hyperparameters, we define $p(\theta)$ as a normal distribution with mean value 0 and $\Sigma_\theta = \sigma^2 I$. Therefore, we can derive the final optimization objective function as below:

$$OF = \ln p(\theta|R) = \ln p(R|\theta) p(\theta)$$
$$= \sum_{u \in U} \sum_{R_u} \ln \sigma(r_{i,j}^u) - \lambda ||\theta||_2^2 \qquad (16)$$

where $r_{i,j}^u = r(u, e_i) - r(u, e_j)$, $\lambda$ is the coefficient of regularization term.

According to the implicit feedback data, we can get the weight vector in the scoring model through maximizing the objective function $OF$.

### 5.2. Optimization Method

Because the optimization objective function is differentiable, we can use the gradient ascent method to estimate the weight vector. The gradient in every iteration be calculate as below:

$$\frac{\partial OF}{\partial \theta} = \frac{\partial}{\partial \theta} \left( \sum_{u \in U} \sum_{R_u} \ln \sigma(r_{i,j}^u) - \lambda ||\theta||_2^2 \right)$$
$$= \sum_{u \in U} \sum_{R_u} \frac{\partial}{\partial \theta} \ln \sigma(r_{i,j}^u) - \lambda \frac{\partial}{\partial \theta} ||\theta||_2^2 \quad (17)$$
$$\propto \sum_{u \in U} \sum_{R_u} \frac{e^{-r_{i,j}^u}}{1 + e^{-r_{i,j}^u}} \cdot \frac{\partial}{\partial \theta} r_{i,j}^u - \lambda \theta$$

From the Eq.(17), it can be seen that there is too much calculation in every iteration of the standard gradient ascend, so we employ the SGD (Stochastic Gradient Descent) method to do the estimation. In every iteration of the learning process, only two user-event pairs of one user are randomly extracted from the training set to update the weight vector, as Eq. (18):

$$\theta = \theta + \alpha \left( \frac{e^{-r_{i,j}^u}}{1 + e^{-r_{i,j}^u}} \cdot \frac{\partial}{\partial \theta} r_{i,j}^u - \lambda \theta \right) \qquad (18)$$

where $\alpha$ is the learning rate, which controls the convergence speed of the learning process. Through Eq.(18), the weight vector with the training data and the hyperparameters ($\alpha$ and $\lambda$) can be estimated. $\alpha$

is often set empirically, and the value of $\lambda$ can affect the effect of the model learning, which will be determined through experiment.

## 6. Experiments

### 6.1. Experiment Design

#### 6.1.1. Dataset Partition

Before carrying out experiments, the dataset is divided into the training set and the test set. The training set is used to learn the weight vector in the recommendation model and the results of our approach are evaluated on the test set. Considering the timeliness of event, we employ a better way to divide the dataset rather than the traditional cross-validation method. The set of the records which are users participated in events are divided into three parts with proportion of 8:1:1 based on event time, denoted as $S_1$, $S_2$ and $S_3$. We define the process of training and test as follows:

1) Training process. The base network of training is constructed on $S_1$, which the computation of meta-path eigenvalue in the training process is based on. For every users, the events with records in $S_2$ constitute his positive feedback set (with value 1), and the events without records in $S_1$ and $S_2$ constitute the negative feedback set (with value 0). We extract training samples from the positive and negative feedback sets and use the model learning method to estimate the weight vector.

2) Test process. The base network of test is constructed on $S_1$ and $S_2$ , which the computation of meta-path eigenvalue in the test process is based on. For users in $S_3$, we use related recommendation methods to carry out Top-N recommendation for them and compute the evaluation metrics through comparing the recommendation results with the user-event records in $S_3$.

#### 6.1.2. Evaluation Metrics

In this paper, we choose the metrics of precision and recall which are often used in recommendation research to evaluate the effectiveness of our recommendation method. The metric of precision is de-

fined as Eq.(19), and the recall metric is defined as Eq.(20).

$$Precision = \frac{\sum\limits_{u \in U_{test}} |R(u) \cap T(u)|}{\sum\limits_{u \in U_{test}} |R(u)|} \quad (19)$$

$$Recall = \frac{\sum\limits_{u \in U_{test}} |R(u) \cap T(u)|}{\sum\limits_{u \in U_{test}} |T(u)|} \quad (20)$$

where $U_{test}$ is the user set in the test set $S_3$, $R(u)$ represents the set of recommendation events for user $u$, $T(u)$ represents the set of events with records in $S_3$ for user $u$.

### 6.1.3. Comparison Methods

To validate the effectiveness of our approach, we compare the proposed method with several typical recommendation methods. The methods involved in the experiment are described as follows:

**MHF:** The **M**ultiple **H**ybrid **F**eatures based event recommendation method proposed in this paper, where the computation of meta-path eigenvalue uses Eq.(3).

**MHF-L:** Also the method proposed in this paper, with the computation of meta-path eigenvalue uses Eq.(4).

**UCF:** The **U**ser based **C**ollaborative **F**iltering event recommendation method. The similarity between users is computed according to the event attending records and the recommendation is carried out based on the similarity using the collaborative filtering method.

**ICF:** The **I**tem based **C**ollaborative **F**iltering event recommendation method. The similarity between events is computed according to the event attending records and the recommendation is carried out based on the similarity using the collaborative filtering method.

**CEER:** The event recommendation method proposed in work 19, which uses the information of content, social relation and local popularity to recommend.

**BMF:** The method which uses the **M**atrix **F**actorization model with **BPR**[36] learning method to recommend.

For the above methods, the UCF and ICF belong to heuristic methods, which do not need the training process. But for the others, there are parameters which need to be learned in the model. And all the learning processes of them have two hyperparameters to be set up, which are the learning rate $\alpha$ and the regularization term coefficient $\lambda$. Learning rate $\alpha$ will be selected by balancing global convergence with speed of convergence. For our methods MHF and MHF-L, the hyperparameters $\lambda$ are determined by experiments. We extract 800 users from the test set $S_3$ as the validation set to carry out recommendation experiments with different $\lambda$ and compute the precision metric of the results, shown in Fig.9.

According to the experiment results, the hyperparameters in our approached are set as follows: For the New York dataset, MHF($\alpha = 0.01, \lambda = 10^{-6}$), MHF-L($\alpha = 0.03, \lambda = 10^{-5}$). For the San Francisco dataset, MHF($\alpha = 0.03, \lambda = 10^{-7}$), MHF-L($\alpha = 0.03, \lambda = 10^{-7}$). And for CEER and BMF, the hyperparameters are set according to their paper as follows: CEER($\alpha = 0.0001, \lambda = 10^{-6}$), BMF($\alpha = 0.001, \lambda = 10^{-3}$) and the size of dimensions for the latent factors in BMF is set to be 10.

### 6.2. Experiment Results

#### 6.2.1. Analysis on the Overall Effect

We carry out experiments of Top-N event recommendation using the above methods on the two city datasets and compare the performance on the metrics of precision and recall, shown in Fig.10.

From the experiment results, it can be seen that our approach performs better than the other typical methods for different values of N on different city dataset. In particular, the BMF method obtain the worst results compared to other methods, we argues that it is caused by the data sparsity which lead to the distortion of feature vectors in the matrix decomposition process. Our method has an obvious higher promotion compared to BMF, which proves that in the case where the data is extremely sparse, making more use of various indirect relations and context information in the network can effectively improve the performance of recommendation.

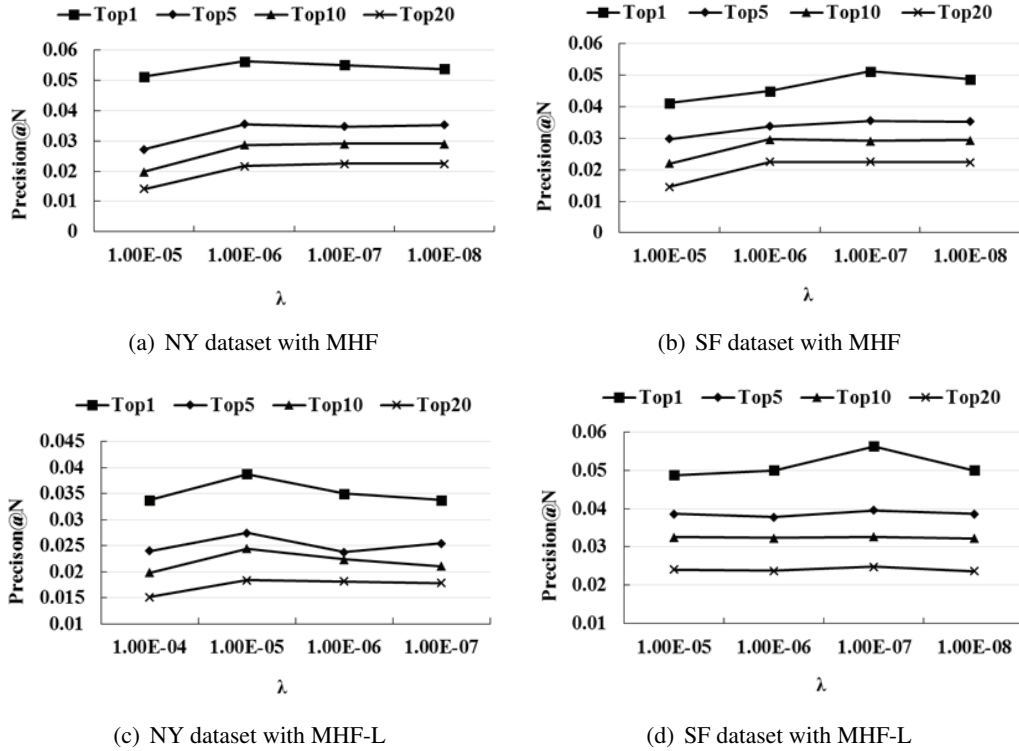For the two methods proposed in this paper MHF

(a) NY dataset with MHF

(b) SF dataset with MHF

(c) NY dataset with MHF-L

(d) SF dataset with MHF-L

Figure 9: Precision Comparison with different $\lambda$

and MHF-L, we can see that the performance of MHF is better than MHF-L. For the computing method of meta-path eigenvalue in MHF-L, it is a way of enhancing the weak connection according to the weak connection theory in essence, but it doesn't work well in EBSN. We argues that it is caused by the strong data sparsity of EBSN. In the case where the data is extremely sparse, the indirect relations can play a great role in recommendation, and these relations can be seen as strong connections in the situation, so it is more appropriate to use the information of the connections without modification, just like Eq.(3).

### 6.2.2. Analysis on the Effect of Different Sparsity

In recommendation problem, the data sparsity of users has direct impact on the effect of recommendation. So we define user's data sparsity according to the number of records between users and events,

shown as follows:

$$Sparsity(u) = \begin{cases} 1, & |E(u)| \leqslant 5 \\ 2, & 6 \leqslant |E(u)| \leqslant 10 \\ 3, & 11 \leqslant |E(u)| \leqslant 15 \\ 4, & 16 \leqslant |E(u)| \leqslant 20 \\ 5, & |E(u)| > 20 \end{cases} \quad (21)$$

where $E(u)$ represents the set of events that user $u$ has attended.

The sparsity from 1 to 5 represents the intensity of the data sparsity from very strong to very weak. In order to validate the effectiveness of the proposed method for users with different sparsity, we carry out experiments of Top-5 recommendation on users with different sparsity, and compare the performance on the metric of precision, shown in Fig.11.

From the experiment results, it can be seen that our proposed event recommendation methods works better than other comparison methods for users with
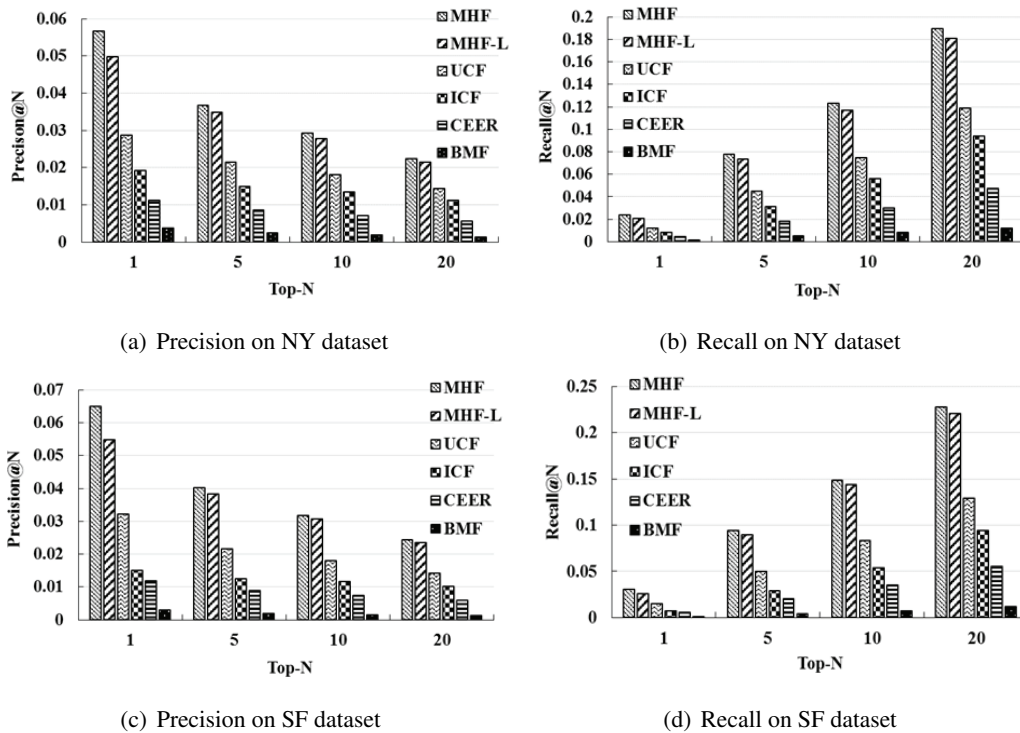
(a) Precision on NY dataset

(b) Recall on NY dataset



(c) Precision on SF dataset

(d) Recall on SF dataset

Figure 10: Precision and Recall Top-N Recommendation Comparison with different Method



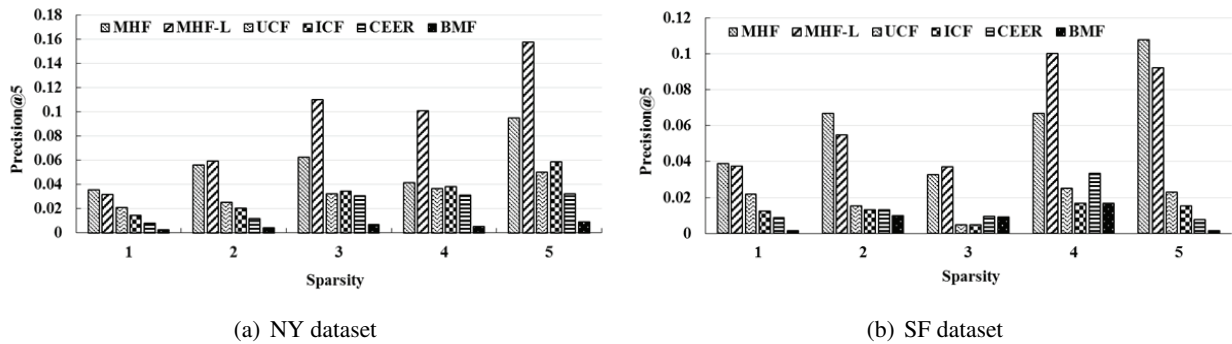(a) NY dataset

(b) SF dataset

Figure 11: Precision Comparison on the Effect with Different Sparsity

different sparsity. In the case where the data is relative abundant, the promotion of our approach compared to other methods is more obvious, and in the case where the data is relative sparse, the effect of our approach also has considerable improvement compared to the others. This result proves that the dealing method of the data sparsity in our approach is effective and the proposed event recommendation algorithm has robustness to the data with different sparsity.

For the two methods proposed in this paper MHF and MHF-L, when the data is relatively sparse, MHF works better than MHF-L, which is just the opposite when the data is relatively abundant. The result is just in line with the analysis in the last section. That is for users with sparse data, the limited links which can be seen as strong connections play an important role to the recommendation, so we should use the

indirect link information without modification and the MHF method which works just in this way performs better. But for the users with abundant data, there are more indirect links available and the difference between the intensity of relations is obvious. So we can construct user's preference more comprehensively through mining the available information in the weak connections in the way of enhancing their effect like Eq.(4) and the MHF-L method which works in this way obtains the better recommendation results.

## 7. Conclusion

In this paper, we propose a multiple hybrid features based method to solve the event recommendation problem. On the basis of constructed the EBSN model, features based on the information of EBSN network structure and context are extracted to measure user's preference on event. We create a recommendation model on the features and use the implicit feedback data to learn the weight vector in the model. Experiments show that our approach has better performance on real datasets compared to other typical methods, which proves that adding the information of indirect relations and context into the recommendation method can alleviate the impact of the data sparsity in EBSN. Although our approach can easily generate online recommendation list by feature weights based on offline training, we will modify the recommendation method to make it suitable for the online real-time network on streaming data[38] environment in the future.

## Acknowledgements

## References

1. Y. Zheng, Location-based social networks: Users, *Computing with spatial trajectories*, (Springer, 2011), pp. 243–276.
2. B. Yu and H.Fei, Modeling social cascade in the flickr social network, in *Proc. 6th Int. Conf. Fuzzy Systems and Knowledge Discovery*, (IEEE, 2009), pp. 566–570.
3. X. Liu, Q. He and, Y. Tian, W. Lee, J. McPherson and J. Han, Event-based social networks: linking the online and offline social worlds, in *Proc. 18th Int. Conf. Knowledge discovery and data mining*, (ACM, 2012), pp. 1032–1040.
4. Z. Wang, L. Sun, W. Zhu, S. Yang, H. Li and D. Wu, Joint social and content recommendation for user-generated videos in online social network, *J. IEEE Transactions on Multimedia*.**15** (3) (2013) 698–709.
5. L. ano-Sanchez, J. A. Recio-Garcia, B. Diaz-Agudo and G. Jimenez-Diaz, Social factors in group recommender systems, *J. Transactions on Intelligent Systems and Technology*. **4** (1) (ACM, 2013) 8.
6. ş. ş. Erçcetin, N. B. Neyişsci, Social Network Analysis: A Brief Introduction to the Theory, *J.Chaos, Complexity and Leadership*, (Springer, 2016), pp. 167–171.
7. J. Naruchitparames, M. H. Gunes and S. J. Louis, Friend recommendations in social networks using genetic algorithms and network topology, in *Proc. of Congress of Evolutionary Computation*, (IEEE, 2011), pp. 2207–2214.
8. P. D. Meo, K. Musial-Gabrys, D. Rosaci, G. M. Sarn and L. Aroyo, Using centrality measures to predict helpfulness-based reputation in trust networks. *Transactions on Internet Technology*, **17**(1)(2017).
9. M.C. Pham, Y. Cao, R. Klamma and M. Jarke, A clustering approach for collaborative filtering recommendation using social network analysis, *Journal of Universal Computer Science*. **17**(4)(2011),583–604.
10. L. Backstrom, D. Huttenlocher, J. Kleinberg and X. Y. Lan, Group formation in large social networks: membership, growth, and evolution, in *Proc. of 12th Int. Conf. on Knowledge discovery and data mining*, (ACM, 2006) 44–54.
11. P. De Meo, E. Ferrara,D. Rosaci and G. M. Sarn, Trust and compactness in social network groups. *IEEE transactions on cybernetics*. **45**(2)(2015), 205–216.
12. D. Rosaci, Finding semantic associations in hierarchically structured groups of Web data, *Formal Aspects of Computing*. **27**(5-6)(2015), 867–884.
13. T. Gruber, Collective knowledge systems: Where the

social web meets the semantic web. *Web semantics: science, services and agents on the World Wide Web.* **6**(1)(2008), 4–13.

14. P. De Meo, A Nocera, D. Rosaci, Domenico and D Ursino, Recommendation of reliable users, social networks and high-quality resources in a social internetworking system, *Ai Communications.* **24**(1)(2011), 31–50.

15. W. Zhang, J.Wang and W. Feng, Combining latent factor model with location features for event-based group recommendation, *Pro. 19th Int. on Knowledge discovery and data mining*, (ACM, 2013), pp. 910–918.

16. Z. Qiao, P. Zhang, Y. Cao, C. Zhou and L. Guo, B, Combining Heterogenous Social and Geographical Information for Event Recommendation, in *Pro. of the conf. on Artificial Intelligence*, (2014), pp. 145–151.

17. X. Ji, Z. Qiao, M. Xu, P. Zhang, C. Zhou and L. Guo, Online Event Recommendation for Event-based Social Networks, in *Proc. of the 24th Int. Conf. on World Wide Web*, (ACM,2015), pp. 45–46.

18. W. Zhang and J. Wang, A collective bayesian poisson factorization model for cold-start local event recommendation, *Proc. of the 21st Int. Conf. on Knowledge Discovery and Data Mining*, (ACM, 2015), pp. 1455–1464.

19. Z. Wang, P. He, L. Shou, K. Chen and S. Wu, Toward the new item problem: context-enhanced event recommendation in event-based social networks, *European Conference on Information Retrieval*, (Springer, 2015), pp. 333–338.

20. A. Q. Macedo, L. B. Marinho and R. L. Santos, Context-aware event recommendation in event-based social networks, in *Proc. of the 9th Int. Conf. on Recommender Systems*, (ACM, 2015), pp. 123–130.

21. T. A. N. Pham, X. Li, G. Cong and Z. Zhang, A general graph-based model for recommendation in event-based social networks, in *Proc. of IEEE 31st Int. Conf. on Data Engineering*, (IEEE, 2015), pp. 567–578.

22. X. Liu, Y Tian, M. Ye and W. C. Lee, Exploring personal impact for group recommendation, in *Proc. of the 21st ACM Int. Conf. on Information and knowledge management*, (ACM, 2012), pp. 674–683.

23. S. Purushotham and C. C. J. Kuo, Modeling group dynamics for personalized group-event recommendation, in *Proc. of Int. Conf. on Social Computing, Behavioral-Cultural Modeling*, (Springer, 2015), pp. 405–411.

24. K. Feng, G. Cong , S. S. Bhowmick, S. Ma, In search of influential event organizers in online social networks, in *Proc. of Int. Conf. on Management of data*, (ACM, 2014), pp. 63–74.

25. R. Du, Z. Yu, T. Mei, Z. Wang, Z. Wang, B. Guo, Predicting activity attendance in event-based social networks: Content, context and social influence, in *Proc. of the 2014 ACM Int. Joint Conf. on Pervasive and Ubiquitous Computing*, (ACM, 2014), pp. 425–434.

26. X. Zhang, J. Zhao and G. Cao, Who Will Attend?–Predicting Event Attendance in Event-Based Social Network, in *proc. of 16th Int. Conf. on Mobile Data Management* , (IEEE, 2011), pp. 74–83.

27. K. Li, W. Lu, S Bhagat , L. V. Lakshmanan and C. Yu, On social event organization, in *Proc. of the 20th Int. Conf. on Knowledge discovery and data mining* , (ACM, 2014), pp. 1206–1215.

28. Y. Tong, R. Meng and J. She, On bottleneck-aware arrangement for event-based social networks, in *Proc. 31st Int. Conf. on Data Engineering Workshops*, (IEEE, 2011), pp. 216–223.

29. D. H. Lee, Pittcult: trust-based cultural event recommenders, in *Proc. of the 2008 ACM Conf. on Recommender systems*, (ACM, 2008), pp. 311–314.

30. E. Minkov, B. Charrow , J. Ledlie and S. Teller, Collaborative future event recommendation, in *Proc. of the 19th Int. Conf. on Information and knowledge management*, (ACM, 2010), pp. 819–828.

31. G. Liao, Y. Zhao, S. Xie and S. Philip, An effective latent networks fusion based model for event recommendation in offline ephemeral social networks, in *Proc. of the 22nd ACM Int. Conf. on information and knowledge management*, (ACM, 2013), pp. 1655–1660.

32. Y. Sun, J. Han, X. Yan, P. S. Yu and T. Wu, Pathsim: Meta path-based top-k similarity search in heterogeneous information networks, in *Proc. of the VLDB Endowment*, (Citeseer, 2011), pp. 992–1003.

33. H. Liwei, L. Deyi, M. Yutao, A meta path-based link prediction model for heterogeneous information networks, *J.Chinese Journal of Computers.* **37** (4) (2008) 848–858.

34. D. Mok and B. Wellman, Does distance matter in the age of the Internet?, *J.Urban Studies.* **47**(13)(2010),2747–2783.

35. J. Cao and Y. D. Zhang, A method of adaptively selecting best LDA model based on density, *Computing with spatial trajectories*, (Springer, 2011), pp. 243–276.

36. S. Rendle, C. Freudenthaler, Z. Gantner and L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, in *Proc. of the 25th Conf. on uncertainty in artificial intelligence*, (AUAI Press, 2009) 452–461.

37. R. Pan, H. Y. Zhou, B. Cao, N. Liu, R. Lukose, M. Scholz and Q. Yang, One-class collaborative filtering, in *Proc. Int. Conf. on Data Mining*, (IEEE, 2008) 502–511.

38. Y. Wang, H. Li, H Wang, B. Zhou and Y.C. Zhang, Multi-window based ensemble learning for classification of imbalanced streaming data, in *Proc. Int. Conf. on Web Information Systems Engineering*, (Springer, 2015) 78–92