Exploiting Social Influence for Context-Aware Event Recommendation in Event-based Social Networks

Zhibo Wang[†,‡], Yongquan Zhang[†], Yijie Li[†], Qian Wang[†,¶,*] and Feng Xia[§]
[†]State Key Lab of Software Engineering, School of Computer, Wuhan University, P. R. China
[‡]Key Lab of Aerospace Information Security and Trusted Computing, Wuhan University, P. R. China
[¶]Collaborative Innovation Center of Geospatial Technology, Wuhan University, P. R. China
[§]School of Software, Dalian University of Technology, P. R. China
Email: {zbwang, dellenzhang, yuizxs00, qianwang}@whu.edu.cn, f.xia@ieee.org

***Abstract*—Event-based Social Networks (EBSNs) which bridge the gap between online and offline interactions among users have received increasing popularity. The unique cold-start nature makes event recommendation more challenging than traditional recommendation problems, since even for two events with the same content, they may not happen at the same time, the same location, or be organized by the same host. Existing event recommendation algorithms mainly exploit the basic context information (e.g., location, time and content), while the social influence of event hosts and group members have been ignored. In this paper, we propose a Social Information Augmented Recommender System (SIARS), which fully exploits the social influence of event hosts and group members together with basic context information for event recommendation. In particular, we combine the information of EBSNs and other social networks to characterize the social influence of event hosts, and take interactions between group members into consideration for event recommendation. In addition, we propose a new content-aware recommendation model using the topic model to find the most similar topic the event belongs to, and a new location-aware recommendation model integrating location popularity with location distribution for event recommendation. Extensive experiments on real-world datasets demonstrate that SIARS outperforms other recommendation algorithms.**

I. Introduction

With the rapid development of social networking services (e.g., Facebook [1], Twitter [2], Google+ [3]), a special kind of social networks called event-based social networks (EBSNs), such as Meetup [4] and Eventbrite [5], have emerged and received popularity in recent years. The EBSNs allow users to create events online and participate the events offline, which provide a new way for users to establish and enhance social ties. So far, EBSNs have attracted a plethora of users and have been undergoing business blooming. According to the statistics of Meetup [6], it has 26 million registered users from 182 countries and more than 580,000 events are organized per month.

The core goal of EBSNs is to gather neighbors (e.g., in the same city) together to do what they are commonly interested in [7]. However, the large volume of events makes it difficult for

*Corresponding author.

Fig. 1. An example of an event on Meetup with five key elements: location, time, attendee, host, and content.

users to find the events they like. For example, about $20,000$ new events are created every day in Meetup [6] and this number is still growing. As a consequence, personalized event recommendation is urgently required to solve the information overload problem and recommend the most interested events to users quickly and efficiently.

The events published in EBSNs could be a technical conference, a house party, a football game, a weekend hiking and so on. Figure 1 shows an example of an event on Meetup which contains several key elements: location, time, attendee, content and host. It is the "event" that shifts users' social interaction from virtually online to physically offline and differentiates the EBSNs from conventional online social networks. However, the unique features of events in EBSNs also make the event recommendation problem more challenging as it faces a more severe cold-start problem. First, the newly published events are scheduled in a near feature, having little or no historical attendance. Second, each event has a life cycle which requires the event recommendation to be made only after the creating time and before the starting time of the event. Moreover, even for two events having the same content, they may not happen at the same time, the same location or be organized by the same event host. In addition, the user feedback information (e.g., ratings and comments) commonly used by classic recommender systems becomes useless in EBSNs, since an event has become over and invalid when the ratings and comments are given by the event attendees after they attended the event. Thus, the unique cold-start nature makes event

978-1-5090-5336-0/17/$31.00 ©2017 IEEE

recommendation problem in EBSNs very challenging, and the classic recommendation algorithms (e.g., collaborative filtering and matrix factorization) cannot be directly applied for event recommendation in EBSNs.

Recently several event recommendation algorithms have been proposed in EBSNs by exploiting location features [8], contextual signals [9], social and geographical information [10], and the information of social relation, event content and event host/organizer [7]. However, the social influence of event hosts and group members are ignored in the existing event recommendation algorithms, which actually play an important role for event recommendation. Although an event is purely new to users, its event host may already have organized several events before, which actually provides us a lot of side information for us to solve the cold-start problem for event recommendation. Moreover, event hosts publishing events in EBSNs usually will also promote the events on their other social networks (e.g., Twitter and Facebook), which makes their social influence an important factor for event recommendation. Although [7] and [11] also used event host information for recommendation, they did not exploit event host's social information. Yin et al. analyzed more than 758 million user pairs' interaction history, and found out that 87.5% of users' co-attend are followed by the same kind of interaction [12]. This implies that the interactions of group members also affect event recommendation since users tend to follow other's choice to attend events if they have co-attended events before.

In this paper, we focus on the personalized event recommendation problem in EBSNs aiming to recommend the most related events to users. To solve the severe cold-start problem in event recommendation, the social influence of event hosts and users' group members are exploited together with contextual information (e.g., location, time and content). We propose a Social Information Augmented Recommender System (SIARS) unifying all the recommendation models including the host-aware, member-aware, time-aware, location-aware and content-aware recommendation model, to calculate the overall recommendation score between any user-event pair. The top-N events with the highest recommendation score will be recommended to the user. The main contributions of this work are summarized as follows:

- To the best of our knowledge, we are the first to exploit the social influence of event hosts and group members for event recommendation.
- We make the first attempt to combine the information of EBSNs and other social networks to characterize the social influence of event hosts, and propose a host-aware recommendation model and a member-aware recommendation model to reflect the social influence for event recommendation.
- We propose a new content-aware recommendation model using the topic model to find the most similar topic the event belongs to, and a new location-aware recommendation model integrating location popularity with location distribution for event recommendation.
- We propose a Social Information Augmented Recommender System (SIARS) unifying all the recommendation models together to calculate the overall recommendation score between any user-event pair.
- We implement and evaluate SIARS with real-world datasets from Meetup, and compare it with state-of-the-art recommendation algorithms. Experimental results show that our algorithm outperforms existing algorithms and improves the accuracy of recommendation.

The rest of the paper is organized as follows. We discuss the related work in Section II and describe the event recommendation problem in Section III. We present the SIARS in Section IV and conduct extensive experiments in Section V. Finally, we conclude the paper in Section VI.

II. Related Work

In this section, we first briefly introduce the location-based recommendation in LBSNs and then discuss the work of event recommendation in EBSNs.

A. Location-based Recommendation in LBSNs

Location-based recommendation has been extensively studied in LBSNs. After the strong correlation was found between geographical distance and social connections in [13] [14], a lot of works focused on leveraging the geographical property for location-based recommendation. Ye et al. [15] observed that users tend to visit locations close to their homes or offices and also may be interested in exploring the nearby places of their visited locations, so they utilized the power-law distribution to exploit the geographical influence of locations for recommendation. Whereas Cheng et al. [16] argued that users tend to visit locations around centers (i.e., the most popular locations) and adopted the multi-center Gaussian model to form the distribution of the distance between a visited location and its center. In [17], Zhang et al. assumed that different users have different location preferences, thus showing different location distribution. Based on that intuition, they proposed a kernel density estimation approach for location-based recommendation. In EBSNs, location is also an important element of events. As was pointed out in [18], events in EBSNs, which need all attendees to meet at the same spot, must be located in locations that are close to most of attendees and can provide convenient services. Thus, the idea of location-based recommendation algorithm can be integrated with other elements of events for event recommendation in EBSNs. In this paper, we propose a new location-aware recommendation model integrating location popularity with location distribution to characterize the influence of location to event recommendation.

B. Event Recommendation in EBSNs

Recently there are a few works studying event recommendation in EBSNs. Liu et al. [18] firstly defined the event-based social network (EBSN) as a co-existence of both online and offline social interactions, and conducted event recommendation using information flow model which only utilized the topological structure of EBSNs for event participation prediction. Afterwards, Qiao et al. [10] proposed a Bayesian matrix factorization approach to event recommendation by exploiting heterogeneous social and geographical information. This approach was later extended by Du et al. [11], who integrated content, location and time into the recommendation algorithm. However, Macedo et al. [19] showed that pure matrix factorization based only on user-event interactions performs poorly on EBSN data in comparison to simpler methods due to the high level of sparsity of these datasets. Recently, Zhang et al. [7] proposed a collective bayesian poisson factorization model to integrate event content, organizer, location and user social relation to infer user latent factors for recommendation. However, the social influence of event hosts and group members have not been exploited for event recommendation. Different from existing event recommendation algorithms, we exploit the social influence of event hosts and group members together with the influence of content, location and time to solve the cold-start problem for event recommendation.

III. System Model And Problem Statement

In this section, we first introduce the system model of EBSNs, and then describe the event recommendation problem.

A. System Model

An EBSN is comprised of three types of entities: users, events and groups. Users can create groups to gather people with common interests online and launch events in groups to allow people to interact with each other offline. A user can be a participant or a host of events. For an event, users can response to events with RSVP* ("yes", "no" or "maybe"). Note that each event only belongs to one group, and users can attend an event even when they do not join the group that the event belongs to.

Figure 1 shows an example of a typical event in EBSNs. Formally, an event consists of five key elements: 1) content, which gives a brief introduction of the event; 2) event host, the user who organizes the event; 3) attendee, who will go to attend the event for a future event or who have attended the event for a past event; 4) location, where the event will be held; 5) time, when the event will be held. It is intuitive to utilize these features to conduct event recommendation.

B. Problem Statement

The event recommendation problem is more challenging in EBSNs due to the unique cold-start nature of events. First, the newly published events are scheduled in a near feature, having little or no historical attendee. Second, each event has a life cycle which requires the event recommendation to be made only after the creating time and before the starting time of the event. Moreover, even for two events having the same content, they may not happen at the same time, the same location or be organized by the same event host. In addition, the user feedback information (e.g., ratings and comments) commonly used by classic recommender systems becomes useless in EBSNs, since an event has become over and invalid when the ratings and comments are given by the event attendees after they attended the event.

*RSVP is a term in meetup.com, means "please respond".

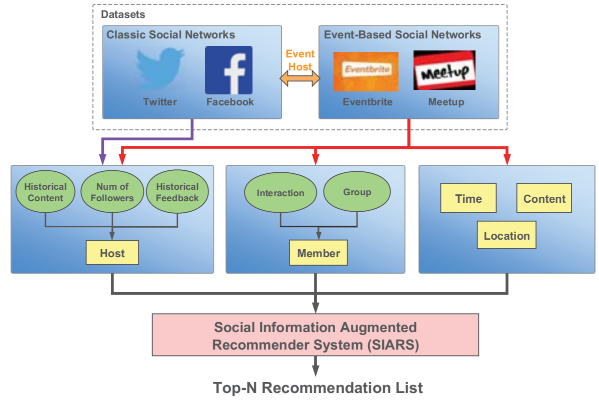

Fig. 2. The high-level overview of the SIARS

***Problem 1** (**Cold-start Event Recommendation**):* Given an EBSN with events, groups and users, the goal is to propose a recommender system to calculate the recommendation score $S(u, e)$ between a user u and a newly created event $e \in E$, where E is the set of newly created events that have not happened yet. Finally, the top-N ranked events in E are recommended to each user.

IV. Social Information Augmented Recommender System (SIARS)

In this section, we present the Social Information Augmented Recommender System (SIARS) to solve the cold-start event recommendation problem. We first give a high-level overview of the system, and then present the recommendation models exploiting different features of events and finally unify these recommendation models together with a fusion model.

A. System Overview

Figure 2 shows the high-level overview of the SIARS. We utilize the datasets from EBSNs and classic online social networks, and exploit the social influence of event hosts and group members together with the influence of basic context (e.g., time, content, and location) for recommendation.

With the datasets from EBSNs (e.g., Meetup, Eventbrite) and conventional social networks (e.g., Twitter, Facebook), we exploit the social influence of event host and propose a host-aware recommendation model that uses historical content, number of followers, and historical feedback of the event host for event recommendation. We adopt the framework of the adaptive boosting personalized ranking algorithm (AdaBPR) and modify it to learn the weights to these three factors. We also propose the member-aware recommendation model which exploits the interactions of user's group members for event recommendation.

Among the five elements of an event shown in Figure 1, location, time and content have been extensively studied for recommendation in LBSNs and EBSNs [8] [10] [15]. In this

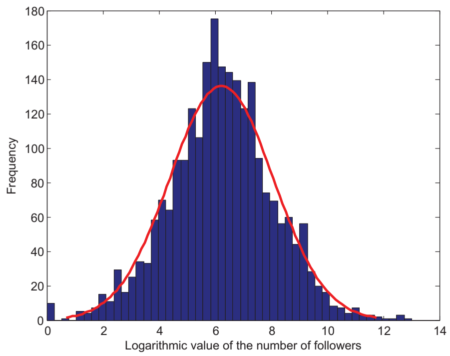

Fig. 3. Social Account Analysis of Event Hosts

section, we regard them as the basic contextual recommendation model. We adopt the extensively studied time-aware recommendation model from [9] and propose new location-aware and content-aware recommendation models considering the new features of EBSNs. In particular, we use the topic model [20] to discover the interested topics from attended events of a user and propose a new content-aware recommendation model to find the most similar topic the event belongs to. We also propose a new location-aware model integrating location popularity with location distribution for event recommendation.

Finally, we use the fusion model of SIARS to unify the five recommendation models together to calculate the overall recommendation score for any user-event pair. The top-N events with the highest scores to the user will be recommended.

B. Social Influence of Event Host

Event host is one of the key elements of events in EBSNs, which has not received too much attention for event recommendation. Although an event is purely new to users, its event host may already have organized several events before, which actually provides us a lot of side information for us to solve the cold-start problem for event recommendation. The event host can be characterized from several aspects. First, as we mentioned, event hosts usually leverage their social accounts on other social networks (e.g., Facebook, Twitter) to advertise the events. That is, event hosts leverage their social influence on other social networks to affect user's attendance. Second, although a new event does not have any feedback from attendees, its host has a lot of historical feedback on the events he/she has hosted, which can be used to characterize the reputation of the event host for event recommendation. Moreover, the quality of the content can be predicted by the historical contents of events hosted by the event host. Therefore, we characterize the social influence of an event host from these three aspects and propose a host-aware recommendation model combining these three heterogeneous social influences together to calculate the preference of users to events.

1) Social Influence on Other Social Networks: It is common that event hosts advertise the event they are going to host on online social networks, like Twitter or Facebook. Generally speaking, the larger social influence of an event host, the more popularity of the event. In Meetup, users can provide their social accounts in their profiles, such as Facebook or Twitter accounts, which can be used to analyze their social influence.

We crawled the social accounts information from Twitter for the event hosts in Meetup. Generally speaking, a user's influence can be judged by the number of followers to he/she as more followers means that more people will know the events. Therefore, we use the number of followers to characterize the social influence of event hosts. Figure 3 shows the histogram of the number of followers on a *log* scale. We can see that the distribution follows almost the Gaussian distribution. Thus, we utilize the cumulative distribution function (CDF) to model the social influence of an event host in terms of social accounts on classic social networks:

$$SA(h) = \int_{-\infty}^{\log(h_d+1)} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx \tag{1}$$

where $SA(h)$ denotes the social influence of the event host h, h_d stands for the number of followers to h. Parameters μ and σ are the mean and standard deviation of the distribution of the number of followers on a *log* scale of the whole dataset. We use $\log(h_d + 1)$ instead of $\log(h_d)$ to avoid infinity.

2) Historical Feedback to The Events Hosted by The Event Host: Due to the cold-start nature of an event, event ratings become useless since an event has become over and invalid when the rating are given by attendees. As a result, classical recommendation algorithms fail to exploit this user feedback information for event recommendation. However, the event ratings to the events is an important metric that can be used to characterize the reputation of event hosts. Generally speaking, the higher the event ratings given by attendees, the higher the reputation of the event host. Here we use the average rating to characterize the social influence of the event host in terms of historical rating.

$$HF(h) = \frac{\sum_{i=1}^{n_h} \sum_{j=1}^{m_i} r_{ij}}{\sum_{i=1}^{n_h} m_i} \tag{2}$$

where $HF(h)$ denotes the social influence of the event host h in terms of historical rating, n_h is the number of events hosted by the event host h, m_i is the number of ratings for the ith event, and r_{ij} is the rating of the jth attendee to the ith event.

3) Historical Contents of Events Hosted by The Event Host: In order to make a good recommendation, we prefer recommending the events hosted by experienced hosts. In other words, it is reasonable to believe that an event whose host has hosted similar events before is more likely to be a good event. This fact commonly exist in real life. For example, an expert in some area is more likely to give excellent talks and a flagship conference is more likely to accept high-selective papers. Therefore, we use the similarity between the event content and the historical events' content to measure the social influence of the event host in terms of event content.

For the content of an event, we represent it using the bag-of-word model and use the TF-IDF weighting of each word to form the content vector of the event.

$$HC(h) = \sum_{e_i \in E_h} \frac{1}{|E_h|} cos(\overrightarrow{e_i}, \overrightarrow{e}) \quad (3)$$

where $HC(h)$ denotes the social influence of the event host h in terms of historical content. E_h is the set of events hosted by the event host h and $\overrightarrow{e_i}$ is the content vector of event i. $\overrightarrow{e}$ is the content vector of the new event e. We utilize cosine similarity measure to calculate the similarity between contents of events.

Host-aware Recommendation Model: With the three recommendation components of an event host, we propose a host-aware recommendation model to linearly combine them together to calculate the recommendation score of a user u to an event e in terms of event host, denoted by $S_h(u, e)$.

$$S_h(u, e) = \lambda_{SA} SA(h) + \lambda_{HF} HF(h) + \lambda_{HC} HC(h) \quad (4)$$

where the three recommendation components $SA(h)$, $HF(h)$ and $HC(h)$ are treated as three ranking functions and λ_{SA}, λ_{HF} and λ_{HC} are the weights for these three ranking functions respectively.

In order to find the appropriate values for the weights of recommendation components, we adopt the framework of an adaptive boosting personalized ranking algorithm (AdaBPR) [21]. However, the recommendation components in AdaBPR are homogeneous derived from the latent factor model, while the three recommendation components of the event host are heterogeneous. Therefore, we adopt the framework of the AdaBPR algorithm and modify it to calculate the weights for the heterogeneous recommendation components in terms of event host in our recommendation model.

Before we elaborate on how to learn the weights for the three ranking functions, we first give some definitions. Let U and H denote the set of users and event hosts, and (u, h) denote a user-host pair. Let H_u^+ represent the set of event hosts who user u has interacted with, and $H_u^- = H \backslash H_u^+$. Given a ranking function $\pi(\cdot)$, π_{uh} is the rank position of event host h to user u. We choose the widely used ranking metric, AUC, to measure the accuracy of the ranking function.

$$AUC(\pi(\cdot)) = \frac{1}{|U|} \sum_{u \in U} \frac{1}{|H_u^+|} \sum_{h \in {H_u}^+} \frac{1}{|H_u^-|} \sum_{h' \in H_u^-} I(\pi_{uh} < \pi_{uh'}) \quad (5)$$

where $I(\cdot)$ is the indicator function, which equals 1 when the condition is true, and 0 otherwise. That is, for a pair of two event hosts (h, h'), since $h \in {H_u}^+$ and $h' \in {H_u}^-$, the host h should be higher than the h' at the ranking list, so $\pi_{uh} < \pi_{uh'}$ and $I(\pi_{uh} < \pi_{uh'}) = 1$. However, if h' is ranked higher than h by $\pi(\cdot)$ at the ranking list, namely $\pi_{uh'} < \pi_{uh}$, so we have $I(\pi_{uh} < \pi_{uh'}) = 0$.

The basic idea of learning the weights is to check each ranking function one by one and adjust the corresponding weight according to the AUC of each ranking function and the weight of each training user-host pair. During the learning process, user-host pairs that was ranked incorrectly by the first ranking function, will be given higher weights for the next ranking function. This in return can increase the weight for this ranking function that ranks them correctly.

For a user u, we let w_{uh} denote the weight for a user-host pair (u, h) and initialize it with $1/|H_u^+|$. For each user u, we choose the ranking functions one by one to rank all the events in terms of event host and compute their ranking accuracy to adjust the corresponding weights.

We first choose the ranking function $SA(\cdot)$ to rank all the events in terms of event host, and obtain the ranking accuracy $AUC(SA(\cdot))$. Then we adjust the weight λ_{SA} for $SA(\cdot)$ with the ranking accuracy $AUC(SA(\cdot))$ and the weights for user-host pairs w_{uh}.

$$\lambda_{SA} = \frac{1}{2} \ln \frac{\sum_{(u,h) \in D} w_{uh} \{1 + AUC(SA(h))\}}{\sum_{(u,h) \in D} w_{uh} \{1 - AUC(SA(h))\}} \quad (6)$$

where D stands for the set of user-host pairs.

After that, we follow the reassigning rule in [21] to reassign weights for the user-host pairs for the second ranking function $HF(\cdot)$. That is,

$$w_{uh}^{HF} = \frac{\frac{1}{|H_u^+|} exp\{-AUC(SA(h))\}}{\sum_{(u,h) \in D} \frac{1}{|H_u^+|} exp\{-AUC(SA(h))\}} \quad (7)$$

Similar to $SA(\cdot)$, we adjust the weight of $HF(\cdot)$ according to its ranking accuracy $AUC(HF(h))$ and the new weights for user-host pairs ω_{uh}^{HF}.

$$\lambda_{HF} = \frac{1}{2} \ln \frac{\sum_{(u,h) \in D} w_{uh}^{HF} \{1 + AUC(HF(h))\}}{\sum_{(u,h) \in D} w_{uh}^{HF} \{1 - AUC(HF(h))\}} \quad (8)$$

Then we reassign weights for the user-host pairs for the third ranking function $HC(\cdot)$.

$$w_{uh}^{HC} = \frac{\frac{1}{|H_u^+|} exp\{-AUC(HF(h))\}}{\sum_{(u,h) \in D} \frac{1}{|H_u^+|} exp\{-AUC(HF(h))\}} \quad (9)$$

At last, we can adjust the weight λ_{HC} for $HC(\cdot)$.

$$\lambda_{HC} = \frac{1}{2} \ln \frac{\sum_{(u,h) \in D} w_{uh}^{HC} \{1 + AUC(HC(h))\}}{\sum_{(u,h) \in D} w_{uh}^{HC} \{1 - AUC(HC(h))\}} \quad (10)$$

With these calculated weights and the three ranking functions (recommendation components), we can calculate the recommendation score in terms of event host with the host-aware recommendation model in Eq. 4.

C. Member-aware Recommendation Model

In [12], Yin et al. analyzed more than 758 million user pairs' interaction history, and concluded that 87.5 percent of users' co-attend are followed by the same kind of interaction. This implies that users' attendance can be affected by others and it is highly possible that two users attend the same event if they have co-attended events before. As we mentioned, an event e is created in a group G_e where members share commonly interest, so the members in this group are usually more interested in this event than members in other groups. Thus, we can characterize the preference of a user u to the event e by checking the co-attendance of this user with all the members in the group G_e. Intuitively, the higher probability of co-attendance with group members, the higher preference to the event.

Let E_u and $E_{u'}$ denote the set of events users u and u' having attended, respectively. We use $sim(E_u, E_{u'})$ to denote the preference similarity of users u and u', which also reflects the influence of user u' to user u on attending events. The Jaccard similarity coefficient is adopted here for calculating $sim(E_u, E_{u'})$ as it measures the similarity between finite sample sets. That is, $sim(E_u, E_{u'}) = \frac{|E_u \cap E_{u'}|}{|E_u \cup E_{u'}|}$. Therefore, the member-aware recommendation model is shown as follows.

$$S_m(u,e) = \frac{1}{|G_e|-1} \sum_{u' \in G_e} sim(E_u, E_{u'}) \tag{11}$$

where $|G_e|$ is the number of members in the group G_e.

D. Content-aware Recommendation Model

Content of an event is obviously one of the important factors affecting users' attendance decision. The content of an event is usually under one topic, so the attended events of a user can reflect his/her interested topics. If the topic of an event is similar to a user's interested topics, it can be recommended to the user. With this intuition, we use the topic model to discover the interested topics of users from their attended events and then present the content-aware recommendation model.

We use the classic topic model, Latent Dirichlent Allocation (LDA) algorithm [20], to derive a user's interested topics from the events he/she has attended. As mentioned earlier, for the content of an event, we represent it using the bag-of-word model and use the TF-IDF weighting of each word to form the content vector of the event. Each content vector of an event can be treated as a document and all the content vectors/documents corresponding to E_u form a corpus for user u, where E_u is the set of events user u has attended. Let d denote a document in the corpus and w_n is the nth word in this document. The probability of this document can be expressed as follows:

$$p(d|\alpha,\beta) = p(\overrightarrow{\theta}|\alpha) \prod_{n=1}^{N} p(w_n|\overrightarrow{\phi}_{z_n}) p(z_n|\overrightarrow{\theta}) p(\overrightarrow{\phi}|\beta) \tag{12}$$

where $\overrightarrow{\theta}$ is the topic distribution parameterized by Dirichlet hyper parameter α, z_n is the nth topic, and $\overrightarrow{\phi}_{z_n}$ is the topic-word distribution of topic z_n parameterized by a Dirichlet hyper parameter β. In particular, $\overrightarrow{\theta} = (p_1, p_2, \cdots)$ where p_i represents the probability of the ith topic. With the LDA algorithm, we can derive the topic distribution $\overrightarrow{\theta}$ and the topic-word distribution for each topic, say $\overrightarrow{\phi}_{z_n}$ for topic z_n.

For a newly created event e, we can also calculate its content vector $\overrightarrow{e}$, which is also the word distribution of the event. We then find the most similar topic of the user u to the event e by using the cosine similarity and use it as the recommendation score in terms of content. That is,

$$S_c(u,e) = \max(\cos(\overrightarrow{e}, \overrightarrow{\phi}_{z_1}), \cdots, \cos(\overrightarrow{e}, \overrightarrow{\phi}_{z_K})) \tag{13}$$

where $\overrightarrow{\phi}_{z_i}$ is the word distribution under the ith topic.

E. Time-aware Recommendation Model

The time of an event is another important factor that might affect users' attendance decision since users have different time schedules. We adopt the time-aware recommendation model from [9], where the authors argued that users attending events in the past at certain days of the week and at certain hours of the day will likely attend events with similar temporal profile in the future.

A user's time profile can be characterized by the events he/she has attended. We use a 24×7-dimensional vector $\overrightarrow{e_t}$ to represent an event e, where an entry is set to 1 when the event occurred at that particular day and hour and 0 otherwise. Thus, the time profile of a user u, denoted by $\overrightarrow{u_t}$, is calculated as the centroid of the events he has attended.

$$\overrightarrow{u_t} := \frac{1}{|E_u|} \sum_{e \in E_u} \overrightarrow{e_t} \tag{14}$$

The time-aware recommendation score $S_t(u,e)$ can be expressed as the cosine similarity between user's time profile vector and the event's time vector, which is

$$S_t(u,e) = \cos(\overrightarrow{u_t}, \overrightarrow{e_t}) \tag{15}$$

F. Location-aware Recommendation Model

In [9], the authors proposed a kernel-based density estimation approach that uses location distribution for location-aware recommendation, while the location popularity has been ignored. In real life, some locations are much more popular than others due to some reasons, such as beautiful scenes or wonderful services. The higher popularity means the better attraction to users. Therefore, we integrate the location popularity into the location-aware model proposed in [9].

The popularity of a location l, denoted by P_l, can be characterized by the number of events held at this location. The more events held at the location, the more popularity it is. Thus, we can represent the popularity P_l as $P_l = \frac{\ln|E_l|}{1+\ln|E_l|}$, where E_l is the set of events held at location l. Note that P_l is a normalized increasing function of $|E_l|$. By integrating the location popularity into the location-aware recommendation model in [9], we have

$$S_l(u,e) = \frac{\ln|E_{l_e}|}{1+\ln|E_{l_e}|} \hat{f}(l_e) \tag{16}$$

where $S_l(u,e)$ is the location-aware recommendation score of event e to user u, l_e is the location of the event e which is a lat-long coordinate, and $\hat{f}(l_e)$ is the location-aware recommendation model in [9].

In particular, $\hat{f}(l_e)$ is calculated as follows:

$$\hat{f}(l_e) = \frac{1}{|L_u|} \sum_{l' \in L_u} K_{\mathbf{H}}(l_e - l') \tag{17}$$

where L_u is the set of locations of events the user u has attended in the past, and $K_{\mathbf{H}}(\cdot)$ is the bivariate Gaussian kernel,

$$K_{\mathbf{H}}(x) = \frac{1}{\sqrt{2\pi|L_u|}} \epsilon^{-\frac{xx^T}{2\sqrt{\mathbf{H}}}} \tag{18}$$

and $\mathbf{H}$ is a 2×2 symmetric and positive definite matrix that represents the bandwidth defined as $\mathbf{H} = (h_1, h_2) \times \mathbb{I}$. For more details about $\hat{f}(l_e)$ please refer to [9].

G. Fusion Model of SIARS

To unify all the recommendation models together, we use the fusion model as follows to calculate the overall recommendation score of a user-event pair (u, e).

$$S(u, e) = \mathbf{w}^T \mathbf{S} \tag{19}$$

where $\mathbf{S} = [S_t(u, e), S_l(u, e), S_c(u, e), S_h(u, e), S_m(u, e)]$ is the feature vector containing the five recommendation scores and $\mathbf{w} = [w_t, w_l, w_c, w_h, w_m]$ is vector of weights for the five recommendation scores. We follow the idea in [8] [9] using a *learning to rank* approach to learn the vector of weights. The goal is to find the vector of weights that rank events that a user has attended higher than events not. We use the AUC as the optimization criterion and adopt the Coordinate Ascent method [22] to learn the vector of weights. Once the vector of weights is learned, the overall recommendation score between any user-event pair can be calculated. For any user, the top-N events with the highest recommendation scores will be recommended.

V. Experimental Evaluation

In this section, we evaluate the performance of the SIARS on real-world datasets crawled from an EBSN Meetup. We implement the SIARS in Python and compare its performance with state-of-the-art recommendation algorithms.

A. Dataset

We use the dataset crawled from Meetup in [9] and the events occurred in three cities, namely Phoenix, Chicago, and San Jose, from January 2012 to January 2014 are selected. We also crawl the events occurred in New York City from January 2014 to January 2016 as part of our dataset. We check the profiles of event hosts and crawl the social account data from Twitter. In the preprocessing phase of dataset, we only select the users who have attended at least 5 events, and the events with at least 5 attendees, and the groups with more than 20 events for further process. After data preprocessing, the statistics of our dataset are shown in Table I.

TABLE I
Dataset Statistics

# of Events	336146
# of Groups	13666
# of Users	104169
# of Event Hosts	10195
# of Locations	38979
# of RSVPs	808071

B. Setup and Metrics

In order to evaluate the recommendation accuracy of the proposed algorithm, we randomly mark off $x\%$ of events attended for each user as the ground truth. We regard them and real cold-start events as the total set of cold-start events, and then use the recommendation algorithms to recover these

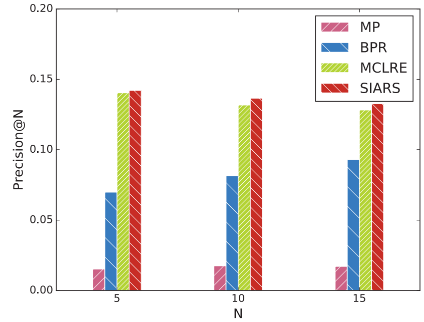

Fig. 4. Comparisons of recommendation algorithms on Precision@N when the mark off rate is 30%

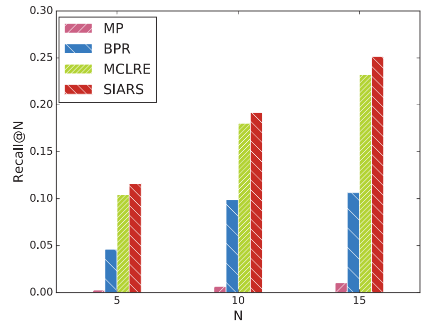

Fig. 5. Comparisons of recommendation algorithms on Recall@N when the mark off rate is 30%

missing user-event pairs that have been marked off. In our experiments, we evaluate the performance when $x = 10$, 30, or 50 with 30 as the default value. Two commonly used metrics are examined to evaluate the recommendation accuracy:

- Precision@N: the ratio of recovered events to the recommended N events.
- Recall@N: the ratio of recovered events to the set of events that have been marked off.

C. Recommendation Baselines

To prove the effectiveness of the SIARS, we compare it with several state-of-the-art recommendation algorithms.

- MP: a baseline which recommends the most popular events.
- BPR: The *Bayesian Personalized Ranking* [23] is a state-of-the-art matrix factorization algorithm for item recommendation from implicit feedback.
- MCLRE: The *Multi-Contextual Learning to Rank Events* [9] is a state-of-the-art context-aware recommendation algorithm that utilizes event's time, location, content and social relation information for event recommendation.

The parameters for all models are tuned by grid search. For MCLRE, we tune its neighbors number in $\{50, 65, 100\}$, time decay parameter in $\{0.005, 0.01, 0.5\}$, and set its bandwidth h_1 and h_2 to 0,001 and 0.00075. For BPR, we set its number of latent factors to 300, learn rate to 0.1, and the number of iterations to 1500.

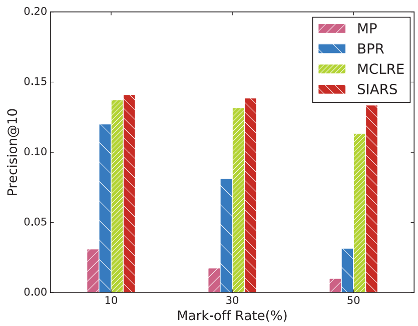

Fig. 6. Comparisons of recommendation algorithms on Precision@10 under different mark off rates

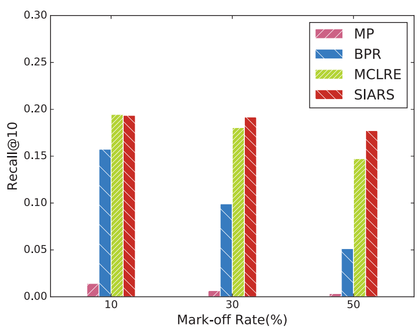

Fig. 7. Comparisons of recommendation algorithms on Recall@10 under different mark off rates

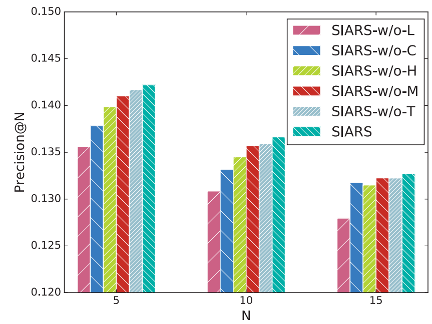

Fig. 8. Comparisons of SIARS with its submethods on Precision@N

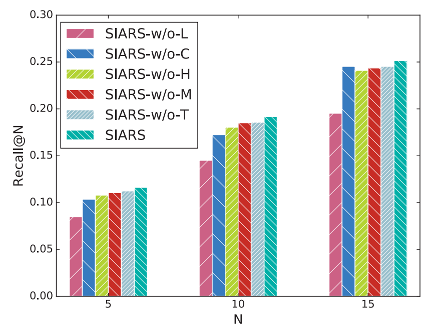

Fig. 9. Comparisons of SIARS with its submethods on Recall@N

D. Recommendation Algorithms Comparison

Figure 4 and Figure 5 show the performance comparison of the recommendation algorithms under different lengths of the recommendation list (N) when the mark off rate is 30%.

We can see that our algorithm achieves the best performance in terms of both precision and recall under all values of N, which shows that the proposed algorithm indeed improves the recommendation accuracy. As a context-aware recommendation algorithm, the MCLRE which achieves the second-best performance. The precision and recall of our algorithm and the MCLRE are much better than MP and the BPR, which validates the idea that context-aware recommendation is more suitable for event recommendation while traditional recommendation algorithms cannot work well for event recommendation due to the unique cold-start nature of events in EBSNs.

Note that since the MCLRE already achieves a good precision and recall, the improvement of our algorithm is not large. However, the performance of our algorithm is still better the MCLRE, which indicates that it is useful to exploit the social influence of event host and group members together with the basic context information for event recommendation.

Figure 6 and Figure 7 show the performance comparison of the recommendation algorithms under different mark off rates when $N = 10$. We can also observe that our algorithm achieves the best precision and recall under all mark off rates. The superiority is more obvious when the mark off rate is large. The context-aware recommendation algorithms achieve much better performance than traditional recommendation algorithms.

E. Feature Analysis

In this section, we further explore the contribution of each contextual feature. We intentionally remove one feature from the SIARS as a submethod of the SIARS and compare their performance. Figure 8 and Figure 9 show the comparison of precision and recall of the SIARS and its submethods. The *SIARS-w/o-L* submethod means that we remove the location-aware recommendation model from the SIARS. Similarly, the *SIARS-w/o-H* submethod means that we remove the host-aware recommendation model from the SIARS.

From Figure 8 and Figure 9, we can see that the precision and recall of the SIARS is better than any of its submethods, which implies that it is better to fully utilize all features for accurate event recommendation. The *SIARS-w/o-L* submethod achieves the worst performance compared to other submethods. This in return implies that the location makes greater contribution than other features to the overall recommendation accuracy. The reason may lie in the fact that the events in EBSNs are location-dependent and users pay more attention to the geographical distance and convenient services of the events when they make decisions. The *SIARS-w/o-C* behaves the second-worst performance among all submethods, which means that the content is the second most important feature that affect the accuracy of event recommendation. Therefore, we can conclude that the location and the content are the two most important factors of events affecting users' decisions on attending the events. This in return will require that each event host pay careful attention to the selection of location and content of an event.

We also observe that the *SIARS-w/o-H* submethod and the *SIARS-w/o-M* submethod achieves worse performance than the *SIARS-w/o-T* submethod, which implies that the host and group members actually plays more important role than the time for event recommendation. This also validate the effectiveness of the idea of combing the social influence of event hosts and group members with the basic context information (e.g, location, content and time) together for event recommendation.

VI. Conclusions

In this paper, we proposed the Social Information Augmented Recommender System (SIARS), which fully exploits the social influence of event hosts and group members together with basic context information for event recommendation. In particular, we combined traditional online social networks with EBSNs to exploit the social influence of event hosts. The interactions between group members were also exploited for event recommendation. In addition, we proposed a new content-aware recommendation model using the topic model to find the most similar topic the event belongs to, and a new location-aware recommendation model integrating location popularity with location distribution for event recommendation.

Extensive experiments on real-world datasets demonstrated that our proposed algorithm outperforms other recommendation algorithms. The experimental results show that traditional recommendation algorithms cannot work well for event recommendation due to the unique cold-start nature of events in EBSNs. We also find that the location and the content are the two most important factors for event recommendation, which requires that each event host pay careful attention to the selection of location and content of an event. Moreover, the social influence of event hosts and group members plays more important role than the time for event recommendation, which validated the effectiveness of the idea of combining the social influence of event hosts and group members with the basic context information (e.g, location, content and time) for accurate event recommendation.

Acknowledgement

This work was supported in part by National Natural Science Foundation of China (Grant No. 61502352), National Basic Research Program of China (Grant No. 2014CB340600), National Natural Science Foundation of China (Grants No. U1636219, 61373167, and 61572106), National High-tech Research and Development Program (Grant No. 2015AA016004, 2015AA016002), Natural Science Foundation of Hubei Province and Jiangsu Province (Grants No. 2015CFB203 and BK20150383), Fundamental Research Funds for the Central Universities (Grants No. 2042015kf0016 and 2042016kf0190), and NSFC-General Technology Foundation Research United Fund (Grant No. U1536204).

References

[1] Facebook, http://www.facebook.com/.
[2] Twitter, http://www.twitter.com/.
[3] Google+, https://plus.google.com/.
[4] Meetup, http://www.meetup.com/.
[5] Eventbrite, http://www.eventbrite.com/.
[6] Meetup statistics, http://www.meetup.com/about/.
[7] W. Zhang and J. Wang, "A collective bayesian poisson factorization model for cold-start local event recommendation," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 1455–1464.
[8] W. Zhang, J. Wang, and W. Feng, "Combining latent factor model with location features for event-based group recommendation," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 910–918.
[9] A. Q. Macedo, L. B. Marinho, and R. L. Santos, "Context-aware event recommendation in event-based social networks," in *Proceedings of the 9th ACM Conference on Recommender Systems*, 2015, pp. 123–130.
[10] Z. Qiao, P. Zhang, Y. Cao, C. Zhou, L. Guo, and B. Fang, "Combining heterogenous social and geographical information for event recommendation," in *Proceedings of Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.
[11] R. Du, Z. Yu, T. Mei, Z. Wang, Z. Wang, and B. Guo, "Predicting activity attendance in event-based social networks: Content, context and social influence," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2014, pp. 425–434.
[12] P. Yin, Q. He, X. Liu, and W.-C. Lee, "It takes two to tango: Exploring social tie development with both online and offline interactions," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 2015.
[13] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 1082–1090.
[14] Z. Cheng, J. Caverlee, K. Lee, and D. Z. Sui, "Exploring millions of footprints in location sharing services." *Proceedings of the 5th International Conference on Weblogs and Social Media*, vol. 2011, pp. 81–88, 2011.
[15] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011, pp. 325–334.
[16] C. Cheng, H. Yang, I. King, and M. R. Lyu, "Fused matrix factorization with geographical and social influence in location-based social networks." in *Proceedings of Twenty-Sixth AAAI Conference on Artificial Intelligence*, vol. 12, 2012, pp. 17–23.
[17] J.-D. Zhang and C.-Y. Chow, "igslr: personalized geo-social location recommendation: a kernel density estimation approach," in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2013, pp. 334–343.
[18] X. Liu, Q. He, Y. Tian, W.-C. Lee, J. McPherson, and J. Han, "Event-based social networks: linking the online and offline social worlds," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 1032–1040.
[19] A. Q. De Macedo and L. B. Marinho, "Event recommendation in event-based social networks." in *HT (Doctoral Consortium/Late-breaking Results/Workshops)*, 2014.
[20] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.
[21] Y. Liu, P. Zhao, A. Sun, and C. Miao, "A boosting algorithm for item recommendation with implicit feedback," in *Proceedings of the 25 international joint conference on artificial intelligence*, vol. 15, 2015, pp. 1792–1798.
[22] D. Metzler and W. B. Croft, "Linear feature-based models for information retrieval," *Information Retrieval*, vol. 10, no. 3, pp. 257–274, 2007.
[23] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, 2009, pp. 452–461.