2016 IEEE 16th International Conference on Data Mining

Whether this participant will attract you to this event? Exploiting Participant Influence for Event Recommendation

Yi Liao, Xinshi Lin, Wai Lam
Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong
Hong Kong
Email: {yliao,xslin,wlam}@se.cuhk.edu.hk

***Abstract*—When a user is making a decision on whether to participate an event in Event-based Social Networks (EBSN), one of the common considerations is who have agreed to join this event. The reason is that existing participants of the event affect the decision of the user, to which we refer as participant influence. However, participant influence is not well studied by previous works. In this paper, we propose an event recommendation model which considers participant influence, exploiting the influence of existing participants, on the decisions of new participants. Specifically, we investigate participant influence in relation to several commonly used contextual aspects of the event based on Poisson factorization. We have conducted extensive experiments on some datasets extracted from a real-world EBSN. The results demonstrate that the consideration of participant influence can improve event recommendation.**

I. Introduction

Event-based social networks (EBSNs), such as Meetup.com and Douban.com, link people's online interactions to offline activities. An event organizer can publish an offline event on an EBSN, calling for participants to join in this event. Usually, the published event owns a homepage on the EBSN, showing the basic information about the event and a list of RSVPs of existing participants. The RSVP information indicates who have already shown the willingness to participate the events. An event recommendation system is essential to help a user discover events that he/she is probably interested in. Typically, the event recommendation problem involves lots of information, providing potentially useful clues to recommender systems for finding events that a user is most likely interested in. Several models [10], [3], [4], [11], [5], [12] have been proposed to solve the problem of event recommendation. Information utilized by these models can be grouped into two types. The first type is the context of the event, such as the text description of the event, the location where the event will be held, and the time when the event will be held. The context of the event forms the basic information of the event. The second type is the social relationship of the users. Usually, an EBSN enables users to build social relations with each other, via group memberships or friendships which can be exploited due to the observation that friends tend to have similar interest.

However, most existing works ignore the RSVPs that are listed on the homepage of the events. Often, when a user browses an event, he/she not only checks whether the basic context information of the event, such as description, location, and time, is interesting and convenient, but also considers whether the existing participants, who have decided to participate the event, are good companions or not. We refer to such consideration as participant influence. This is a rather common phenomenon because an offline event may involve lots of interactions between participants. In some cases, existing participants may affect the decisions of the users more than the basic context information of the events. For example, when a user decides to participate a party, some particular existing participants may attract him/her, rather than the basic information.

The notion of participant influence is different from various social relations among users which have been studied in previous works [4], [11], [12], [10]. Social relations are utilized by existing works as auxiliary information, which conveys the message that users sharing the social relations tend to have similar interest. For example, Zhang and Wang [11] utilize friendships between users to infer interest of each user more precisely, based on the assumption that friends tend to have similar interest. Interestingly, they also report that the method they make use of friendship only improves event recommendation performance slightly. In the framework proposed by Macedo et al. [4], users are linked via the groups to which they belong. Such information is utilized to facilitate recommendation assuming that users affiliated to the same or similar groups are prone to attending the same event created by these groups. Xu et al. [10] introduce the concept of mutual influence to investigate which members of a group will participate the event once the event is published. Their model assumes that users strengthen the likelihood of each other to participate the event via the mutual influence, which is modeled as a weighted link. However, the above three models still rely on the basic information of the events for calculating the preference of the user, with social relations acting as auxiliary information. In contrast, the participant

The work described in this paper is substantially supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: 14203414) and the Direct Grant of the Faculty of Engineering, CUHK (Project Code: 4055034). This work is also affiliated with the CUHK MoE-Microsoft Key Laboratory of Human-centric Computing and Interface Technologies.

2374-8486/16 $31.00 © 2016 IEEE
DOI 10.1109/ICDM.2016.149

influence proposed in this paper can be viewed as a latent link between users that will directly affect the decisions of a new participant. Indeed, whether a user will participate an event does not merely depend on the preference, especially for users who care more about who will also join, than what the event is.

We study the problem of event recommendation in EBSNs considering the participant influence. We observe from real-world EBSNs that the proposed participant influence has two characteristics. First, the influence of an existing participant on a new participant exhibits variations under different situations. Motivated by this observation, our proposed framework investigates participant influence regarding different aspects of the event. Specifically, the aspects that we have taken into consideration include location, time, group, and topics of the event. The second characteristic is that some users tend to easily get influenced by other participants, while some do not. Likewise, some users are more likely to influence new potential participants than others. Hence, our framework investigates how a user influences others and gets influenced by others considering the above observations. We model the strength of participant influence with the aforementioned elements. Then we employ Poisson distribution [9], [1] to model the real-valued count that is associated with participant influence. Poisson factorization has the merit of handling the sparsity issue. The model learns influence representation for each user, and the representation of event aspects, which are utilized for recommending events.

II. Problem Definition

The problem we investigate can be regarded as a kind of event recommendation. The problem involves the following elements: event set E, user set U, location set L, group set G. Each event $e \in E$ is associated with basic information, including event location $l \in L$, event time slot $t \in T$, event group $g \in G$, and textual description of the event. In many EBSNs, users sharing similar interest form a group and events can be published under the group. Consequently, each event is always associated with a group. Each event e is also associated with a list of RSVPs, indicating who has expressed his/her willingness to participate the event and the time of the RSVP. All the users can view the basic information and current RSVPs of an event by which they are potentially affected when deciding whether to participate. We refer to such consideration as participant influence.

Similar to the traditional event recommendation task, the aim is to recommend events to a particular user. On top of that, the recommendation model should take into account of participant influence.

III. Model Description

A. Participant Influence Modeling

Our model captures the participant influence of two users regarding several aspects including the location, time, and group of the event. In principle, more aspects can be integrated into our model. It is not effective to design a fully parameterized framework, which includes a specific parameter for each two users and each aspect. First, it suffers from the data sparsity issue. Second, it cannot be generalized to users who have absolutely no interactions. Hence we design a model based on Poisson factorization to capture the participant influence of one user to another. The elements involved in the model, such as locations, groups, time, and so on, are represented by latent factors. The idea of latent factor model is to model the participant influence captured by the observation $N_{u,v}$, with latent factors which are learned in the training stage. With latent factor model, the participant influence between any two users can be predicted. As mentioned in [1], Poisson factorization is a form of probabilistic matrix factorization that replaces the usual Gaussian likelihood with Poisson likelihood. Compared with Gaussian factorization, Poisson factorization is more efficient in inference and it handles sparse data better. We describe our framework in detail below.

We decompose the full participant influence regarding one aspect into K latent influence facets. For each user, we design two latent factors, represented as K-dimensional vectors, in correspondence with the K latent influence facets. Each dimension in the first factor captures the strength that the user influences others for the corresponding facet, denoted as latent active influence $\theta^a_{u,1:K}$. Similarly, each dimension in the second factor captures the likelihood that the user is influenced by others, denoted as latent passive influence $\theta^p_{u,1:K}$. Such design captures the intuition regarding the likelihood that users influence others and get influenced by others are not exactly the same. Besides the factors associated with users, we also design factors for events. Specifically, we design a latent factor for each location denoted as θ_l, a latent factor for each time slot denoted as θ_t, a latent factor for each group denoted as θ_g, and a latent factor for each word in the vocabulary for the textual descriptions of events denoted as θ_w. The above factors are also K-dimensional vectors. Each dimension captures the degree that participant influence between two users for this particular facet will add to the full participant influence. For example, a location corresponding to a college and a location corresponding to a shopping mall may affect participant influence differently.

The full participant influence is jointly decided by two user factors and aspect factors, forming a three-dimensional factor. Let us consider the location aspect. Assuming the user v participated the event before the user u did, one entry in the tensor represents the strength of participant influence, i.e. $PI^l_{u,v}$ for the user v on the user u, regarding the location l. In other words, it denotes how likely the participation of the user v will result in the participation of the user u, when the event is held at the location l.

Normally modeling the observed elements in a tensor requires tensor factorization. However, since the tensor in our case is sparse, we can simplify the tensor factorization as pair-wise interaction of the involved factors [7], i.e. θ^a_v, θ^p_u and θ_l. Consequently, the strength of participant influence is formulated as follows:

$$\begin{aligned} PI^{l_i}_{u,v} =& \theta^p_u\theta^a_v + \theta_{l_i}\theta^p_u + \theta_{l_i}\theta^a_v \\ =& \sum_{k=1}^{K}(\theta^p_{u,k}\theta^a_{v,k} + \theta_{l_i,k}\theta^p_{u,k} + \theta_{l_i,k}\theta^a_{v,k}) \end{aligned} \tag{1}$$

Similarly, we formulate the participant influence of v on u regarding the time t_j in Equation 2, regarding the group g_m in Equation 3, and regarding the word w_d in Equation 4:

$$PI_{u,v}^{t_j} = \theta_u^p \theta_v^a + \theta_{t_j} \theta_u^p + \theta_{t_j} \theta_v^a \quad (2)$$

$$PI_{u,v}^{g_m} = \theta_u^p \theta_v^a + \theta_{g_m} \theta_u^p + \theta_{g_m} \theta_v^a \quad (3)$$

$$PI_{u,v}^{w_d} = \theta_u^p \theta_v^a + \theta_{w_d} \theta_u^p + \theta_{w_d} \theta_v^a \quad (4)$$

Till now the participant influence is modeled in a matrix factorization form. However, learning a matrix factorization model requires solving a sparse semi-definite programming (SDP), making this approach infeasible for datasets containing millions of observations. Moreover, it is prone to over-fitting. We employ the idea of probabilistic matrix factorization (PMF) [8] to tackle the problem. Specifically, we choose the Poisson distribution to model the observed real-valued count that is associated with the participant influences. For example, the Poisson distribution for a real-valued preference count $N_{u,v}^{l_i}$ for the user v on the user u regarding the location l_i is denoted as:

$$P(N_{u,v}^{l_i}; PI_{u,v}^{l_i}) = (PI_{u,k}^{l_i})^{N_{u,v}^{l_i}} \exp(-PI_{u,k}^{l_i})/(N_{u,v}^{l_i})! \quad (5)$$

where $PI_{u,v}^{l_i}$ refers to the participant influence defined in Equation 1. The participant influence acts as the shape parameters of the Poisson distribution, which governs the generation of the real-valued preference count $N_{u,v}^{l_i}$. We develop a method for obtaining the count as described in Section III-B. We apply a Gamma prior with parameters $\lambda_{a\cdot}$ and $\lambda_{b\cdot}$ to each latent factor described above to avoid over-fitting, which is also conjugate to Poisson distribution.

The generative process is descried as follows:

1) For each user u,
   a) draw latent active influence factor $\theta_v^a \sim Gamma(\lambda_{ua}^a, \lambda_{ub}^a)$
   b) draw latent passive influence factor $\theta_u^p \sim Gamma(\lambda_{ua}^p, \lambda_{ub}^p)$
2) For each location l, draw latent factor $\theta_l \sim Gamma(\lambda_{la}, \lambda_{lb})$
3) For each time, draw latent factor $\theta_t \sim Gamma(\lambda_{ta}, \lambda_{tb})$
4) For each group, draw latent factor $\theta_g \sim Gamma(\lambda_{ga}, \lambda_{gb})$
5) For each word, draw latent factor $\theta_w \sim Gamma(\lambda_{wa}, \lambda_{wb})$
6) For two users (u, v) that the RSVP of v occurs before that of u in an event,
   a) For each location l_i, draw count $N_{u,v}^{l_i} \sim Poisson(PI_{u,v}^{l_i})$
   b) For each time t_j, draw count $N_{u,v}^{t_j} \sim Poisson(PI_{u,v}^{t_j})$
   c) For each group g_m, draw count $N_{u,v}^{g_m} \sim Poisson(PI_{u,v}^{g_m})$
   d) For each word in the vocabulary of textual descriptions of events, draw count $N_{u,v}^{w_d} \sim Poisson(PI_{u,v}^{w_d})$

B. Preference Count Aggregation

This subsection describes a method for acquiring the real-valued preference count associated with participant influence denoted by a basic notation $N_{u,v}$ in our model. For each user-pair (u, v), we collect all the events that the RSVP of the user v occurs before the RSVP of the user u, composing a set denoted as $E_{u,v}$, the size of which is denoted by $N_{u,v}$.

For each aspect, we further aggregate all the events in the set $E_{u,v}$ into clusters, according to the distinct attribute value of the aspect. Then we count the number of instances in each cluster. For example, for the location aspect, we extract n distinct locations and forms the location set $L = \{l_1, l_2, ..., l_n\}$. We aggregate all the events such that the events within the same cluster share the same location. Let $C_{u,v}^{l_i}$ denote the cluster corresponding to the location l_i, which contains all the events in the set $E_{u,v}$ that are held at the location l_i, and let $N_{u,v}^{l_i}$ denote the count of events in the cluster $C_{u,v}^{l_i}$. The count $N_{u,v}^{l_i}$ captures how many times that the user v registers an event held at the location l_i, before the user u does, which can be viewed as an indicator of participant influence Intuitively, a higher $N_{u,v}^{l_i}$ reflects that the user v have higher participant influence on the user u when the location of the event is l_i. We model each aspect and each cluster separately to capture the participant influence under different circumstances. Note that in this setting, an event belong to multiple $E_{u,v}$, as there exist several RSVP pairs in an event. Consequently, an event typically belongs to several clusters that are derived from $E_{u,v}$, since each event is associated with several aspects.

Besides the above aspects, we also investigate another aspect, namely, the topics of event, which is inferred from the text description of event. For the above two users (u, v), we aggregate all the text descriptions of all the events belonging to the set $E_{u,v}$. The aggregated descriptions form the document $D_{u,v}$, conveying the information that what topics of the events have resulted in the participant influence of the user v on the user u. Let $N_{u,v}^{w_d}$ denote the count of the word w_d in the document $D_{u,v}$.

C. Inference Method

The goal of posterior inference is to infer the latent factors $\Theta = \{\theta_v^a, \theta_u^p, \theta_l, \theta_t, \theta_g, \theta_w\}$, given the observations, denoted by $D = \{N_{u,v}^l, N_{u,v}^t, N_{u,v}^g, N_{u,v}^w\}$. We suppress the hyperparameters in the Gamma distributions, namely, λ_a and λ_b, for simplicity. The posterior distribution of latent factors can be expressed as $p(\Theta|D) = \frac{p(D|\Theta)p(\Theta)}{p(D)}$. The denominator is the marginal probability of all the observed data.Unfortunately, the probability is intractable due to the coupling of multiple factors in the integration. Inspired by [11] and [1], we develop an inference method based on variational inference algorithm [2]. The general idea of variational inference is to find a distribution to approximate the intractable posterior distribution. Specifically, a variational distribution $q(\Theta)$ is learned such that the KL divergence to the posterior distribution is minimized.

1) Auxiliary Variables: We add auxiliary variables to facilitate the inference. Note that K denotes the dimension of the latent factors in our model. First, we add K latent variables $\eta_{uv,k} \sim Poisson(\theta_{uk}^a \theta_{vk}^p)$ for each user pairs (u, v). Then we add latent variables for each of the aspect. For example, for the location aspect, we add variables as follows:

1) K latent variables $\alpha_{ul,k}^a \sim Poisson(\theta_{uk}^a \theta_{lk})$ for each user-location pair (u, l).
2) K latent variables $\alpha_{ul,k}^p \sim Poisson(\theta_{uk}^p \theta_{lk})$ for each user-location pair (u, l).

TABLE I
LATENT VARIABLES AND COMPLETE CONDITIONALS

Latent variable	Type	Complete conditional	variational parameters
$\theta^a_{v,k}$	$Gamma$	$\lambda^a_{va} - 1 + \sum_{u \neq v} \eta_{uv,k} + \sum_l \alpha^a_{vl,k} + \sum_t \beta^a_{vt,k} + \sum_g \gamma^a_{vg,k} + \sum_w \delta^a_{vw,k}$ $\lambda^p_{vb} + \sum_{u \neq v} \theta^p_{u,k} + \sum_l \theta_{l,k} + \sum_t \theta_{t,k} + \sum_g \theta_{g,k} + \sum_w \theta_{w,k}$	$\theta^{ashp}_{v,k}, \theta^{arte}_{v,k}$
$\theta^p_{u,k}$	$Gamma$	$\lambda^p_{va} - 1 + \sum_{v \neq u} \eta_{uv,k} + \sum_l \alpha^p_{ul,k} + \sum_t \beta^p_{ut,k} + \sum_g \gamma^p_{ug,k} + \sum_w \delta^p_{uw,k}$ $\lambda^b_{vb} - 1 + \sum_{v \neq u} \theta^a_{u,k} + \sum_l \theta_{l,k} + \sum_t \theta_{t,k} + \sum_g \theta_{g,k} + \sum_w \theta_{w,k}$	$\theta^{pshp}_{u,k}, \theta^{prte}_{u,k}$
$\theta_{l,k}$	$Gamma$	$\lambda_{la} - 1 + \sum_u \sum_v (\alpha^a_{vl,k} + \alpha^p_{ul,k}), \quad \lambda_{lb} + \sum_u \sum_v (\theta^a_{v,k} + \theta^a_{v,k})$	$\theta^{shp}_{l,k}, \theta^{rte}_{l,k}$
$\theta_{t,k}$	$Gamma$	$\lambda_{ta} - 1 + \sum_u \sum_v (\beta^a_{vt,k} + \beta^p_{ut,k}), \quad \lambda_{tb} + \sum_u \sum_v (\theta^a_{v,k} + \theta^a_{u,k})$	$\theta^{shp}_{t,k}, \theta^{rte}_{t,k}$
$\theta_{g,k}$	$Gamma$	$\lambda_{ga} - 1 + \sum_u \sum_v (\gamma^a_{vg,k} + \gamma^p_{ug,k}), \quad \lambda_{gb} + \sum_u \sum_v (\theta^a_{v,k} + \theta^a_{u,k})$	$\theta^{shp}_{g,k}, \theta^{rte}_{g,k}$
$\theta_{w,k}$	$Gamma$	$\lambda_{wa} - 1 + \sum_u \sum_v (\delta^a_{vw,k} + \delta^p_{uw,k}), \quad \lambda_{wb} + \sum_u \sum_v (\theta^a_{v,k} + \theta^a_{u,k})$	$\theta^{shp}_{w,k}, \theta^{rte}_{w,k}$
$\eta_{uv,k}$	$Mult$	$\log \theta^a_{v,k} + \log \theta^p_{u,k}$	$\xi_{uv,k}$
$\alpha^a_{ul,k}$	$Mult$	$\log \theta^a_{u,k} + \log \theta_{l,k}$	$\rho^a_{ul,k}$
$\alpha^p_{ul,k}$	$Mult$	$\log \theta^p_{u,k} + \log \theta_{l,k}$	$\rho^p_{ul,k}$
$\beta^a_{ut,k}$	$Mult$	$\log \theta^a_{u,k} + \log \theta_{t,k}$	$\sigma^a_{ut,k}$
$\beta^p_{ut,k}$	$Mult$	$\log \theta^p_{u,k} + \log \theta_{t,k}$	$\sigma^p_{ut,k}$
$\gamma^a_{ug,k}$	$Mult$	$\log \theta^a_{u,k} + \log \theta_{g,k}$	$\phi^a_{ug,k}$
$\gamma^p_{ug,k}$	$Mult$	$\log \theta^p_{u,k} + \log \theta_{g,k}$	$\phi^p_{ug,k}$
$\delta^a_{uw,k}$	$Mult$	$\log \theta^a_{u,k} + \log \theta_{w,k}$	$\psi^a_{uw,k}$
$\delta^p_{uw,k}$	$Mult$	$\log \theta^p_{u,k} + \log \theta_{w,k}$	$\psi^p_{uw,k}$

which should satisfy the constraint that $N^l_{u,v} = \sum_{k=1}^{K}(\alpha^a_{vl,k} + \alpha^p_{ul,k} + \eta_{uv,k})$. Similarly, we design auxiliary variables and constraint for each of other aspects, including time, group and word, which are denoted by $\beta^{a(p)}_{ut,k}$, $\gamma^{a(p)}_{ug,k}$, and $\delta^{a(p)}_{uw,k}$ respectively. As pointed by Gopalan et al. [1], a sum of independent Poisson random variables is itself a Poisson with the rate equal to the sum of the rates. Hence the new latent variables preserve the marginal distribution of the observed data D.

Then we derive the complete conditional distribution for each latent variable. First, we derive the conditional distribution of original latent variables. For example, let us consider the location factor. The complete conditional distribution for the k-th dimension of the factor $\theta_{l,k}$, given the other latent variables, is formulated as:

$$
\begin{aligned}
&p(\theta_{l,k}|\Theta_{\neg\theta_{l,k}}, \eta, \alpha, \beta, \gamma, \delta, \lambda_a, \lambda_b) \\
&= Gamma(\lambda_{la} - 1 + \sum_u \sum_v (\alpha^a_{al,k} + \alpha^p_{ul,k}), \\
&\lambda_{lb} + \sum_u \sum_v (\theta^a_{v,k} + \theta^a_{v,k}))
\end{aligned} \tag{6}
$$

Then we derive the conditional distribution of auxiliary variables. It is not straightforward to derive the variational factor for the auxiliary factors. Based on the conclusion from [1], we directly show the distribution of auxiliary variables in Table I, including the distribution type and associated parameters. For example, let us consider the location aspect. The Multinomia probability can be interpreted as:

$$
\begin{aligned}
&p(\eta_{uv,k}, \alpha^a_{ul,k}, \alpha^p_{ul,k}|\Theta_{\neg\theta_{l,k}}, \eta, \alpha, \beta, \gamma, \delta, \lambda_a, \lambda_b) \\
=&Mult(\eta_{uv,k}, \alpha^a_{ul,k}, \alpha^p_{ul,k}; \xi_{uv,k}, \rho^a_{ul,k}, \rho^p_{ul,k})
\end{aligned} \tag{7}
$$

We show all the complete conditionals in Table I.

2) Coordinate Updates: We define the mean-field variational family $q(\Theta, \eta, \alpha, \beta, \gamma, \delta)$ where all the variables, including auxiliary variables, are independent variables with distributions shown in Table I. Note that we optimize the variational parameters. Each variational parameter is updated while the other variables are fixed. For example, the update for the variational parameters for location factor $\theta_{l,k}$ is formulated as:

$$
\hat{\theta}^{shp}_{l,k} = E(\lambda_{la} - 1 + \sum_u \sum_v (\alpha^a_{vl,k} + \alpha^p_{ul,k})) \tag{8}
$$

$$
\hat{\theta}^{rte}_{l,k} = E(\lambda_{lb} + \sum_u \sum_v (\theta^a_{v,k} + \theta^p_{u,k})) \tag{9}
$$

where $E(\cdot)$ denotes the expectation. Recall that the value of variables $\alpha^a_{vl,k}$ and $\alpha^p_{ul,k}$ are governed by Multinomial distribution as denoted in Equation 7. $E(\alpha^a_{vl,k}) = N^l_{u,v}\rho^a_{ul,k}$. Regarding the expectation of $\theta^a_{v,k}$ and $\theta^p_{u,k}$, we employ the knowledge that the expectation of Gamma variable is the ratio of the shape parameter and the rate parameter, e.g. $E(\theta^a_{v,k}) = \theta^{ashp}_{v,k}/\theta^{arte}_{v,k}$. By calculating the expectation value for each element, we derive the update form as:

$$
\hat{\theta}^{shp}_{l,k} = \lambda_{la} - 1 + \sum_v N^l_{u,v}\rho^a_{vl,k} + \sum_u N^l_{u,v}\rho^p_{ul,k} \tag{10}
$$

$$
\hat{\theta}^{rte}_{l,k} = \lambda_{lb} + \sum_v \frac{\theta^{ashp}_{v,k}}{\theta^{arte}_{v,k}} + \sum_u \frac{\theta^{pshp}_{u,k}}{\theta^{prte}_{u,k}} \tag{11}
$$

Another type of parameters that we need to updates is the variational parameters of Multinomial variables. Similarly, they are updated via the expectation values. Note that the variational parameters of Multinomial variables are represented by Gamma variables. The expectation of the logarithm of a Gamma variable is, for example, $E(\log \theta^a_{u,k}) = \Psi(\theta^{ashp}_{u,k}) - \log \theta^{arte}_{u,k}$, where $\Psi(\cdot)$ is the digamma function. Hence we update the Multinomial parameters as:

$$
\hat{\eta}_{uv,k} = \exp(\Psi(\theta^{ashp}_{v,k}) + \Psi(\theta^{ashp}_{u,k}) - \log \theta^{arte}_{u,k}\theta^{arte}_{v,k}) \tag{12}
$$

$$
\hat{\alpha}^a_{vl,k} = \exp(\Psi(\theta^{ashp}_{v,k}) + \Psi(\theta^{shp}_{l,k}) - \log \theta^{rte}_{l,k}\theta^{arte}_{v,k}) \tag{13}
$$

$$
\hat{\alpha}^p_{ul,k} = \exp(\Psi(\theta^{pshp}_{u,k}) + \Psi(\theta^{shp}_{l,k}) - \log \theta^{rte}_{l,k}\theta^{prte}_{u,k}) \tag{14}
$$

D. Recommendation Procedure

In the previous stage of our framework, we have learned the location factor, the time factor, and the topic factor, represented by θ_l, θ_t, and θ_g respectively. We further compute θ_e, i.e. the topics of a given event, by averaging the factor of all the words that appear in the text description of the event, denoted as $\theta_e = \frac{1}{|D_e|}\sum_{w \in D_e} \theta_w$, where D_e denotes the words in the text description.

When making recommendation with participant influence, we jointly utilize the location, time, group, and topic of the event. Given a user u and an event e associated with RSVPs, we compute the mean participant influence, denoted by $MPI_{u,e}$. For example, $MPI^l_{u,e}$ is computed as:

$$MPI^l_{u,e} = \frac{1}{|RSVP|}\sum_{v}^{RSVP_e} PI^l_{u,e} \tag{15}$$

We further compute the preference based on the participant influence, denoted as:

$$\begin{aligned} S(u,e) =& E(\lambda_1 MPI^l_{u,e} + \lambda_2 MPI^t_{u,e} + \\ & \lambda_3 MPI^g_{u,e} + \lambda_4 MPI^e_{u,e}) \\ =& \lambda_1 E(MPI^l_{u,e}) + \lambda_2 E(MPI^t_{u,e}) + \\ & \lambda_3 E(MPI^g_{u,e}) + \lambda_4 E(MPI^t_{u,e}) \end{aligned} \tag{16}$$

where $E(\cdot)$ denotes expectation, an example of which is given below:

$$\begin{aligned} & E(MPI^l_{u,e}) = \frac{1}{|RSVP_e|}\sum_{v}^{RSVP_e} E(PI^l_{u,e}) \\ =& \frac{1}{|RSVP_e|}\sum_{v}^{RSVP_e}\sum_{k}^{K}\left(\frac{\theta^{ashp}_{v,k}}{\theta^{ashp}_{v,k}}\frac{\theta^{shp}_{l,k}}{\theta^{shp}_{l,k}} + \frac{\theta^{pshp}_{u,k}}{\theta^{pshp}_{u,k}}\frac{\theta^{shp}_{l,k}}{\theta^{shp}_{l,k}} + \frac{\theta^{ashp}_{v,k}}{\theta^{ashp}_{v,k}}\frac{\theta^{pshp}_{v,k}}{\theta^{pshp}_{u,k}}\right) \end{aligned} \tag{17}$$

We design a method analogously to the BPR optimization criterion [6], to learn the coefficients λs, which emphasize the participant influence regarding different aspects with different weights. To achieve this, we apply a regression model. Given a user e, we compute the preference for all events by Equation 16. Then we select the preference for any event pair, one of which is participated by the user, denoted as $e_{u,y}$ and the other is not, denoted as $e_{u,n}$. We design the regression ranking model as:

$$R(u, e_{u,y}, e_{u,n}) = \frac{1}{1 + \exp(-(S(u, e_{u,y}) - S(u, e_{u,n})))} \tag{18}$$

Then we resolve finding the optimal coefficients λ as maximizing the following $\log$ likelihood function:

$$\mathcal{L} = \log \prod_{u \in U}\prod_{e_{u,y}}\prod_{e_{u,n}} R(u, e_{u,y}, e_{u,n}) \tag{19}$$

The optimization for Equation 19 can be achieved by standard gradient-based method. For recommendation given a user, we compute the score for all the candidate events with Equation 16, incorporating the learned coefficients. Then the candidate events are ranked by the computed scores. Events with higher scores are recommended to the user.

IV. Experiments

A. Experimental Setup

We prepare our raw data from Meetup.com, a popular EBSN website. All the data are acquired via the official Application Programming Interface (API). We have collected the data from four cities, namely, Los Angeles (LA), London (LD), Singapore (SG), and Hong Kong (HK), ranked by the volume of data. Specifically, for each city, we have obtained the events that have been created between 1st, August 2013 and 1st, August 2015, as well as the location and time that each event was held at. A piece of RSVP data represents that a user registers an event. Three elements can be extracted from each RSVP, namely, the user id, the event id, and the timestamp the user registered the event, from which we can identify the order the users registered the event. Note that the organizer of an event is regarded as the first participant of the event. We collected all the RSVPs associated with each event. Events that contain no RSVPs are removed. Since the time of the event is continuous, we define 24 time slots, in correspondence with 24 hours in a day. Each event time is then allocated to one of the time slots. Then each event is associated with a specific time slot. The statistics of the raw data is depicted in Table II.

TABLE II
STATISTICS OF RAW DATA

Number of	LA	LD	SG	HK
Event	228942	109383	26539	18593
User	476782	554044	118875	66136
RSVP	1358494	1066670	253126	143344
Location	26613	19207	5007	3587
Group	7138	7510	2006	1353

We follow the method in [4] in deriving several datasets for our experiments from the above raw data. Since each event has its lifetime, i.e. creation time and occurrence time, the candidate events to be recommended should only include those which have already been created, but not yet occurred. Such setup can facilitate a more realistic simulation of the practical situation. For each city, we derive four datasets named as Part I, II, III, IV. The datasets are created as follows: We select four time points, simulating the situation that the recommender system is trying to recommend events at this time point. We generate a dataset for each time point. The events that are created within 6 months before the time point, as well as the corresponding RSVPs with timestamps before this particular time point, are treated as the training data of the corresponding dataset. We further select events which have been created but not yet occurred at this particular time point, and select the corresponding RSVPs of the events after the time point, to form the data for testing.

To evaluate the performance of event recommendation, we employ normalized discounted cumulative gain (NDCG) evaluation metric truncated to the top 10 recommendations, namely, $NDCG_{10}$. It measures the recommendation quality of an event for a given user based on its position in the recommendation list. In the experiments, we refer to our model as Participant Influence for Event Recommendation (PIER).

B. Comparative Methods

We compare the effectiveness of our proposed event recommendation model (PIER) with three state-of-the-art models. All the three models have been reported to achieve better performance than lots of baselines in event recommendation task. These comparative models are described below.

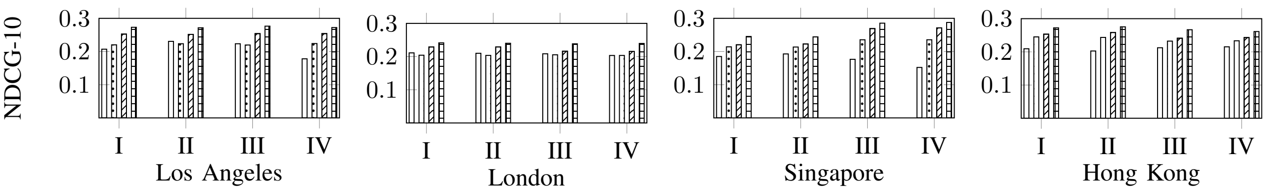

Fig. 1. Comparison between our model and the comparative method. From the left to the right, the bars refer to BMFh, CBPF, MCLRE, PIER respectively.The differences between our model (PIER) and the comparative models are statistically significant for all datasets based on the paired t-test with $p < 0.05$.

Multi-Contextual Learning to Rank (MCLRE) [4] The MCLRE method exploits some contextual information of the users and events. Several features are distilled from the contextual information, which are fed into a learning-to-rank algorithm. Specifically, four types of contextual information, namely, the group information of users and event, the textual description of events, the location of the events, and the time of the event, are utilized to derive the features. Let x denote the feature set for a user-event pair and let y denote whether the user participates the event (1 for yes and 0 for no). The goal is to learn the function $h(x)$ such that the implication: $h(x_i) > h(x_j) \Leftrightarrow y_i > y_j$ holds for any user-event pairs.

Collective Bayesian Poisson Factorization (CBPF) [11] The CBPF method is a collective matrix factorization model which takes Bayesian Poisson factorization as its basic unit to model user response to events, social relation, and content text separately. An event is represented as a weighted combination of organizer, location, and textual information. Given a user and an event, a rating is computed to evaluate the events, based on the multiple types of factors during training. The factor representation of two users tends to become similar if they are friends. Since our dataset does not provide the information of friendship, we alternatively treat two users as friends if they belong to at least one common group.

Bayesian Matrix Factorization with heterogeneous social regularization (BMFh) [5] The BMFh method designs two types of social relations between two users. The first type of relation models the online relation. Specifically, the Jaccard similarity regarding group membership is evaluated to represent the weight between two users. The second relation models the offline relation, i.e. a weight capturing the offline event co-participation. Then Gaussian regularization terms, incorporating these relations, are added to the Bayesian Personalized Ranking (BPR) model [6].

C. Recommendation Effectiveness

We evaluate the effectiveness of PIER for our model, MCLRE, CBPF and BMFh in event recommendation with NDCG-10. The performance of each dataset in each of the cities is shown in Figure 1. It can be observed that our proposed model (PIER) consistently outperforms MCLRE, CBPF, and BMFh in all the datasets, indicating that users are often affected by the existing participants when they decide whether to participate an event. The consideration of existing participants can improve the performance of event recommendation. Moreover, the MCLRE model generally performs better than CBPF and BMFh. Also, CBPF is not worse than BMFh in most cases. The gaps between BMFh and other models are larger in smaller datasets, i.e. Singapore and London. The BMFh is not as stable as other models. One possible reason is that it considers fewer aspects than other models. The CBPF model performs worse than PIER and MCLRE. The reason is that CBPF does not consider group information of the event. In EBSNs, each event is usually associated with a group that holds the event, which should be treated as important information in event recommendation task. Within a city, the result of each model does not show much difference in four datasets, except that in Singapore dataset.

REFERENCES

[1] P. K. Gopalan, L. Charlin, and D. Blei, “Content-based recommendations with poisson factorization,” in *Advances in Neural Information Processing Systems*, 2014, pp. 3176–3184.

[2] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, “An introduction to variational methods for graphical models,” *Machine Learning*, vol. 37, no. 2, pp. 183–233, 1999.

[3] 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.

[4] 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.

[5] Z. Qiao, P. Zhang, C. Zhou, Y. Cao, L. Guo, and Y. Zhang, “Event recommendation in event-based social networks,” in *28th AAAI Conference on Artificial Intelligence*, 2014, pp. 3130–3131.

[6] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “Bpr: Bayesian personalized ranking from implicit feedback,” in *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 452–461.

[7] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, “Factorizing personalized markov chains for next-basket recommendation,” in *Proceedings of the 19th International Conference on World Wide Web*, 2010, pp. 811–820.

[8] R. Salakhutdinov and A. Mnih, “Probabilistic matrix factorization,” in *Advances in Neural Information Processing Systems*, 2008, pp. 1257–1264.

[9] C. K. Vaca, A. Mantrach, A. Jaimes, and M. Saerens, “A time-based collective factorization for topic discovery and monitoring in news,” in *Proceedings of the 23rd International Conference on World Wide Web*, 2014, pp. 527–538.

[10] T. Xu, H. Zhong, H. Zhu, H. Xiong, E. Chen, and G. Liu, “Exploring the impact of dynamic mutual influence on social event participation,” in *Proceedings of the 2015 SIAM International Conference on Data Mining*, 2015, pp. 262–270.

[11] 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.

[12] Y. Zhang, H. Wu, V. Sorathia, and V. K. Prasanna, “Event recommendation in social networks with linked data enablement,” in *15th International Conference on Enterprise Information Systems*, 2013, pp. 371–379.