Personalized Event Recommendations using Social Networks

Ioannis Boutsis, Stavroula Karanikolaou, Vana Kalogeraki
Department of Informatics, Athens University of Economics and Business, Athens, Greece
{mpoutsis, karanikolaou, vana}@aueb.gr

***Abstract*—In recent years we have observed a significant increase in the popularity of location-based social networks for exchanging news and experiences, sharing location information, or publishing real world events. One important challenge in such networks is to understand human crowd mobility behavior based on user social activities and interactions. In this paper we introduce PRESENT, our middleware that utilizes a Mixed Markov Model to extract the behavioral patterns of the users in social groups, to make personalized event recommendations. Our detailed experimental evaluation, using data from the Meetup location-based social network, illustrates that our approach is efficient, practical and achieves an average prediction for the user attendance of over 73%.**

I. Introduction

Recently, the prevalence of social networks along with the widespread adoption of smartphones with ubiquitous sensing capabilities, are driving the development of applications and services that are revolutionizing the way that we interact among each other. Newly emerged location-based social networks such as Foursquare [1], Meetup [2] and Twitter [3], have evolved into attractive platforms for exchanging news and experiences, sharing location information or publishing group activities, *i.e.,* from informal gatherings (*e.g.*, movie nights) to formal activities (*e.g.*, business meetings). Recent studies reveal that an increasing number of people desire to share their geographic location information with their social circles by responding whether they will attend social events (RSVP) or "checking in" at various locations. Such location-based social networks can have significant impact on future businesses as they give the opportunity to companies and organizations to gain insight into human crowd behavior, reach wider communities and improve customer service.

However, this tremendous increase in popularity and size of location-based social networks has led to an information overload. Selecting the most interesting events and deciding whether to attend becomes increasingly challenging, as users have to search over a large number of events published by their social circles to find the ones which are most interesting to them. This procedure is becoming extremely difficult as users become socially connected with several groups. To compensate for the lack of assessment for these events, the user's social network can be used as an additional knowledge source. For example, human crowd behavior is largely influenced by social relationships, as people are more likely to visit places or attend events that people they know of, attend. This is true for business events (*i.e.,* international conferences, workshops), as well as cultural events (*i.e.,* going to movies, concerts). Thus, recommendations of events considering the preferences of members of the social groups are more suitable than traditional recommendations based on the preferences of the individual user. Moreover, social group attendance to events can be very dynamic and depend on several factors such as the group members' availability, time and location of the event, personal interests, etc.

In this paper we develop an approach that exploits user behavior in social groups to make personalized event recommendations. More specifically, the questions we aim to answer are: (1) Can we identify the features that shape human crowd behavior at social events? (2) Given that a user belongs to multiple social groups, can we recommend the next event for a user to attend, based on the features and degree of closeness between members of the social groups? Our goal is to make event attend recommendations that are both effective and personalized. The problem of recommending the next event to a user is challenging due to: (i) the numerous following events that may exist as candidates for each user, (ii) random factors not known to the system may prevent the user to attend some events (*e.g.*, user might be unavailable), (iii) human factor is unpredictable and thus users may decide to attend an event based on several parameters, (iv) users do not respond to all RSVPs, so the attendance lists are incomplete.

Existing event recommendation techniques that utilize metrics including user interests and location preferences [4], popularity and geographical proximity to the event [5], or friendship-based recommendations [6] are inadequate as they fail to capture the behavior developed among users within a social group. Other approaches [7], [8], [9], aim to predict whether a user will attend a forthcoming event or determine his next check-in by focusing on the individual user features, such as past events, categorical preferences, friends attendance, etc. Such approaches face a cold-start problem, especially for short-term events considered in [9], and as we also show in the experimental evaluation, considering user social behavior outperforms recommendation based on individual user features. Other recommendation systems aim to define events (or friends) that the user might be interested to attend (or connect with), but they focus on maximizing the recall of the recommended events rather than the precision [10]. Collaborative approaches that consider previous user behavior [11], [12] as well as combining these approaches with content-based algorithms [13], have also been proposed. However, they focus on evaluating and suggesting each individual event for a user, rather than defining the user's following event attendance. Finally, we state that traditional event recommendation techniques are not adequate for our problem since they do not capture the dynamics developed among social groups, and

Id	Cat. Description	Id	Cat. Description
1	Arts & Culture	12	LGBT
2	Career & Business	13	Movements & Politics
3	Cars & Motorcycle	14	Health & Wellbeing
4	Community & Environment	15	Hobbies & Crafts
5	Dancing	16	Language & Ethnic Identity
6	Education & Learning	17	Lifestyle
8	Fashion & Beauty	18	Literature & Writing
9	Fitness	20	Movies & Film
10	Food & Drink	21	Music
11	Games	22	New & Age & Spirituality

TABLE I. MEETUP CATEGORIES

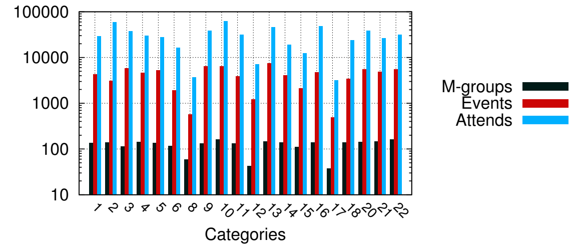

Fig. 1. Meetup Statistics.

they do not consider any sequence in the list of events (*e.g.*, events that overlap temporally will be recommended). These are fundamental differences in our approach.

We summarize our contributions below:

- We introduce PRESENT (PRediction of Event attendance in Social ENvironmenTs), our middleware that aims to exploit event attendance behavior developed within social groups for personalized event recommendations.
- We study several measures of social behavior to quantify the similarity of the user behavior with other members of the social groups. We assume that users develop groups that behave similarly, in terms of a predefined feature, when attending events. PRESENT exploits information from multiple social networks, and utilizes a Mixed Markov Model to extract the behavioral patterns of the groups to determine the next event for a user to attend.
- We have implemented PRESENT and developed a smartphone application that alerts the user for the next event to attend. Our detailed experimental evaluation, using a real location-based social network dataset from Meetup, illustrates that PRESENT is practical, efficient and can predict the next event for a user with high accuracy.

To the best of our knowledge, our work presents the first approach to understand the features that define human crowd behavior at social events, from empirical analysis to prediction models.

II. SYSTEM MODEL

Location-based Social Networks (LBSN). Location-based social networks refer to social networks that incorporate the dimension of geographic location, where users can share location-embedded information or content. Introducing location capabilities, such as geotagging, can reveal interdependencies among users, derived from their locations in the physical world. Such interdependencies can provide knowledge about users with common interests, behavior, etc. Users develop social ties with other members within the groups; these can have significant impact on the structure of the social groups as well as the event attendances within the group. This knowledge can be extracted from social networks, such as Meetup [2], whose purpose is to organize events at different physical locations. For example, Meetup (one of the most popular location-based social networks) enables its members to join Meetup groups, denoted as *M-groups*, where they announce events to their members. Meetup users can also announce their willingness to attend an event for the M-groups that they belong to. M-groups span several categories, as shown in table I. In this work we study the Meetup social network with 205684 users and 89952 events, announced from 2578 M-groups. Figure 1 shows the distribution of groups, events and attends for each Meetup category (more details about our study are provided in the experimental evaluation section).

Social Events. We denote as an event $e \in E$, a pre-organized gathering of people during a particular time interval at a spatial location. Thus, an event can be a football match, a music festival, a session in a conference, etc. Each event e has the form: $< id_e, latitude_e, longitude_e, timestamp_e >$ where id_e is the unique identifier of the event, $latitude_e$ and $longitude_e$ reflect the geographical coordinates where the event takes place, and $timestamp_e$ denotes the starting time of the event. Although in this work we are not interested in the duration of the event, such information can be easily extracted from the Meetup social network.

Users. Each user $n \in U$ can be a member of one or more social networks, and thus, the user can participate in multiple events organized across different social groups. Whenever a user n attends or is willing to attend a social event e, we denote it using a tuple $< id_e, id_n >$ that encapsulates the identifiers of the user and the event. The user attendance to an event can be defined either (i) formally, when the user responds to a formal event invitation (RSVP) and informs the social network about the attendance at event id_e (*i.e.,* Meetup), or (ii) informally, when the user informs the social network about his/her spatiotemporal presence (*i.e.,* using "check-ins" in the case of Foursquare), so that the links among the user's presence and corresponding event can be instantiated [14]. The participation information provides both historical data, from the announced user attendances for completed events, as well as attendances to future events based on user feedback. Thus, such feedback can be the reply to an RSVP for event-based networks, registration forms for formal events, etc. For instance, users tend to register early in conferences or buy tickets online to get lower rates.

III. PROBLEM DEFINITION

Our goal is to better understand the factors that influence human crowd behavior, and in particular, we focus on user behavior in terms of the social groups they attend. We study several features and degree of closeness between members of a social group. Then, we use machine learning techniques, and more specifically prediction techniques, where a set of sample values is first observed and then a statistical model is trained to efficiently predict the next event attendance for a user, when similar preferences are observed. This way, we can better understand user interests, make efficient recommendations about upcoming events as well as identify popular social groups [15]. We give a formal definition of our problem below:

Definition. *Let* $E = e_1, e_2, ..., e_t$ *be the complete set of historic events attended by a set of users* U*, obtained from the users' social profiles. Assuming that users in* U *are organized into* j *groups* G_j*, where each group contains users with similar behavior based on a predefined feature. We consider a set of users* $T \subset U$*, denoted as the training set, who are used to extract the transition probabilities for their group and for whom their willingness to attend future events is known, based on their RSVP responses. Let a set of users* $R \subset U$*, with* $R \cap T = \emptyset$ *and* $R \cup T = U$*, denoted as the evaluated set and a timestamp* t*. Our problem is to rank all the possible next events based on the transition probabilities, to determine the next event* $e \in E$ *for each user* $n \in R$ *to attend after* t*.*

IV. PRESENT MIDDLEWARE

PRESENT uses a graphical model which is a natural way to capture dependencies between variables; these will be used to summarize past-observed events and to predict next event attendance. We exploit the Mixed Markov Model (MMM) [16], which is more efficient compared to the simple Markov Model (MM) and the Hidden Markov Model (HMM) for a number of reasons: MM assumes that all users choose next events to attend on the basis of their current events only. However, this assumption does not consider user personal preferences or social influences. On the other hand, HMM takes into account the user profiles and assumes that users move to different events based on their personal thoughts only. This is inappropriate for our setting as there are various other factors, such as geographical distance or time of the event, that affect the user choices. MMM is an improvement of MM since it takes into account the personalities of the individual users within a group. This characteristic enables MMM to outperform the simple MM and HMM models. MMM studies common attitudes among users in a group. Hence, we can exploit this model as we aim to investigate whether we can predict the user's next attend based on the behavior of the groups he belongs. Finally, non-Markov based approaches like sequence mining and motif mining [17] are not suitable for our setting since each social event is unique and independent, and thus, there are no patterns developed in the sequence that users visit the events.

The architecture of our approach is presented in figure 2. The PRESENT middleware comprises components that run on cloud servers as well as mobile devices to efficiently and accurately provide personalized event recommendations. PRESENT has a **Construction Phase** that resides on a cloud server and consists of the following components: (i) Data Collector, (ii) Model Generator, (iii) User Profiles and (iv) Prediction Component. Additionally a **Suggestion Phase** is executed on the user's mobile devices through the Next Event Suggestion Component. The components are presented in detail in the following sections.

A. Construction Phase

First we describe the construction phase, which is executed on the cloud to instantiate the needed structures for the

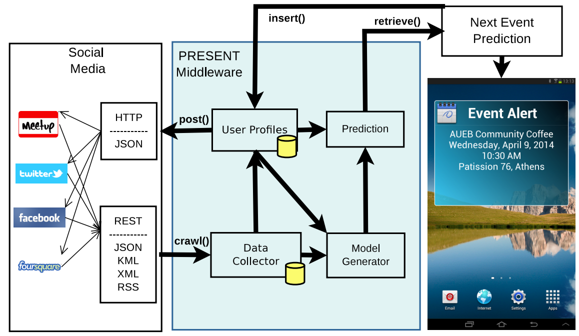

Fig. 2. PRESENT Middleware.

prediction. The construction phase is triggered whenever new data become available from the Data Collector component, to update the structures.

1) Data Collector Component: The Data Collector component is executed periodically to extract new rich content from one or more social networks. Thus, it is executed by calling the `crawl(crawltime, network, parameters[])` function. The crawl function aims to retrieve all the data produced after time `crawltime`, for the selected `network` (*e.g.*, Meetup, Foursquare). The `parameters[]` variable refers to a list of parameters for the requests, such as the type of the data to be retrieved (*e.g.*, events, RSVPs). For instance, we can crawl all Meetup events, produced after timepoint $prevtime$, by calling `crawl(prevtime, Meetup, events)`.

The crawl function is responsible to connect with different social networks, such as Twitter, Foursquare, Meetup, etc., and extract the requested data through their APIs. Each network may support a different protocol for the data retrieval, such as REST or SOAP, as well as different data formats (*e.g.*, JSON, XML, KML, etc.). However, all the social networks that we currently support (Meetup and Foursquare), use the REST API. Thus, they receive a GET message and reply with a JSON object that encapsulates the data. Thus, for an M-group, named "group" the respective call in our previous example would be: $https : //api.meetup.com/2/events/?status = past\&order = time\&group\_urlname = group\&format = json\&fields = \&time = prevtime\&key = 123$". Upon retrieving the JSON object we remove any redundant information and insert the data in a database. We note that the PRESENT middleware can operate with multiple social networks simultaneously.

2) User Profiles Component: The User Profiles component is responsible to maintain user attendances to events. These can be either retrieved from the Data Collector component or from the user's mobile application. When the mobile application sends user attendance information to PRESENT, the `post(user, event, networks[])` function is triggered. This function sends multiple POST requests with JSON formatted objects, for each individual social network (`networks[]`) that the user has connected with. Although each network has different characteristics, PRESENT can post these data asynchronously and transparently to multiple networks. For instance, an event attendance will be represented as a "check-in" in Foursquare and as an RSVP to Meetup.

3) Feature-Based Groups: In order to instantiate our structures for the event recommendation, we divide the users into

$e \in E$	Social Events
$n \in U$	Users in our system
G_j	Groups of users with a similar feature
r	Label for an event (r=1,2,3,...)
m	Amount of states (events)
i	i^{th} state of each user
$w_{n,i}$	n^{th} user's position in each state
$w_{n,i,r} w_{n,i+1,s}$	$i+1$ transition of the n^{th} user to event s from event r
p_{rs}	Transition probability from event r to event s
$p(k)_{rs}$	Transition probability from event r to event s for group k
z	Latent vector for each transition
z_n	Latent vector describing the n^{th} user's group
π_k	Mixing coefficient such that $\pi_k = p(z_{n,k} = 1)$

TABLE II. SYMBOLS AND NOTATIONS

feature-based groups G_j, where their members develop similar behaviors for the selected feature. Our goal is to look for quantities that capture some degree of closeness between members of the groups. Social networks depict different types of features that can be used to organize users into groups based on their behavior such as historical visits, temporal features, categorical preferences and social distance of the visited places [10], [8], [18]. In this work we exploit the most popular features in the literature, namely the spatial and attendance features. This is certainly not an exhaustive set of features, but our study shows that these can greatly affect the attendance of events. Although different features can be plugged into our system, comparing different features is out of the scope of this paper. In our work we consider the following features:

1) **Attendance Criterion**, where we produce groups whose members have similar amount of attends. Assuming a list L with users sorted based on the amount of their attendances, we define the groups as $G_j = \{n | \, \forall n \text{ s.t. } \lfloor L_n/g \rfloor == j\}$, where L_n denotes the sequential position of user n in L and g denotes the maximum group size.
2) **Spatial Criterion**, where we divide the users based on the geographic location of their attendance. Hence, users who visit spatially close places will be assigned to the same group G_j. For each user n, we define a location loc_n, which is the geographical center of the places he has visited. We define the groups G_j by scanning the spatial area, and for each user we discover, we select g users with the smallest distance to loc_n and add them to group G_j.

4) Model Generator Component: The Model Generator Component exploits the data from the Data Collector and User Profiles components, to generate the model we use for the prediction. We employ a graphical representation, as shown in figure 3, to illustrate the mobility of the users of a group. Each chain represents a feature-based group G_j, whose users move among the events (nodes) with some probability that we need to estimate. Such a representation enables us to understand the behavior of the feature based groups G_j. It also enables us to evaluate users under different levels of granularity (*e.g.*, evaluate only high-attendance groups). For instance, we can extract major group events by determining the events attended by a group of users who participate in only a few events.

Events. We use the events attended by the users in groups G_j, extracted through their social profiles, to learn the model. Thus, the set of events to evaluate is denoted as $\{e | \exists < id_e, id_n >, n \in G_j \forall G_j\}$. We only consider the events

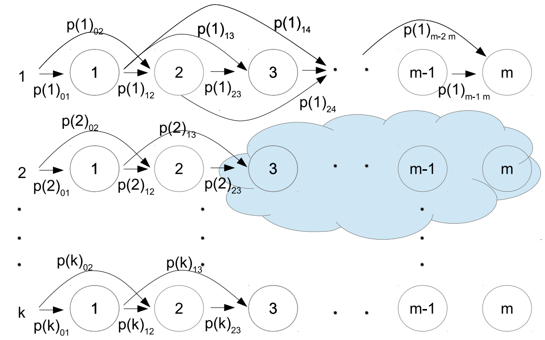

Fig. 3. Transition Probabilities for the Groups.

that have been attended by at least one user in the evaluated groups G_j and so we only need to define whether a user will attend these events. This reduces the computation overhead, compared to the alternative naive approach of linking each user to any produced event. We note that events can overlap for the evaluated groups and that repetitive events are handled as individual events.

Suppose we have a set of observations for each user n that reflects the past events that the user has attended, although the user might have also attended additional events that we are not aware of. We assume that a user moves among the events in the group that we evaluate and we will study each of the groups that he belongs to (in case that the user belongs to more than one group) separately. Also, we assume that at any point in time each user can only move to a future event.

States and Transitions. Initially, we enumerate the events organized by each group with natural numbers $l \in \mathbb{N}$ (l={ 0, 1, 2, ...}) according to the chronological order of the events. Essentially, these represent the complete set of states in our model. A discrete transition of each user is indicated by a series of natural numbers $l \in \mathbb{N}$ depending on which events the user attends. Assume that at the beginning the user is at "zero" event. We use an m-dimensional vector $w_{n,i}$, for each user n, that depends on the amount of events each group has organized in the period we study. Each state i represents the unique event that a specific user attends and m shows all the events e that have been organized by the group in time-series. The vector $w_{n,i} = w_{n,i,1}, ..., w_{n,i,m}$ represents the position of the n^{th} user in each state i, namely the event that the user attends at a particular time instance. There are as many vectors $w_{n,i}$ as the number of events that the specific user has attended. If a group G_j has organized 3 events, then $w_{n,i} = w_{n,i,1}, w_{n,i,2}, w_{n,i,3}$, where $w_{n,i,1}$ is the first event that the n^{th} user has the chance to attend and $w_{n,i,3}$ is the last event he can attend. For instance, if the second event that the user n has visited is the 3^{rd} event of the group then $w_{n,2,3} = 1$ and $w_{n,2,l} = 0, \forall l \in \{1, 2\}$.

Transition Probabilities. In order to predict whether a user $n \in U$ attends an upcoming event, we should compute the probabilities of the user's transitions among the events. We assume that all events are independent since social networks do not capture the similarity or dependancy among events, although such information could further improve our model. We denote the transition probability from event $r \in E$ to event $s \in E$ as p_{rs}, where $\sum_s p_{rs} = 1$, thus the sum of the transition probabilities from event r to every other event equals to one. The probability p_{rs} is related to the transition probabilities among the various events that are organized in a group and so it differs for each different group. We say that $w_{n,i,r}w_{n,i+1,s} = 1$ when the n^{th} user makes the $i+1$ transition to event s on condition that his previous event was r, otherwise $w_{n,i,r}w_{n,i+1,s} = 0$. As a result p_{rs} represents the probability $P(w_{n,i,r}w_{n,i+1,s} = 1)$. We state that we do not consider those $w_{n,i}$ which are equal to zero; we only consider the events that the user n has attended. Therefore the overall transition probability is given as:

$$P(\{w_{n,i,0}, ..., w_{n,i,m}\}) = \prod_{irs} p_{rs}^{w_{n,i,r}w_{n,i+1,s}} \tag{1}$$

Group-based Transition Probabilities. Assuming k groups of users, we present our model for the different groups in figure 3. Note that for simplicity we illustrate only some of the transitions. The states of each chain represent the m events which are depicted in time series. As the figure shows, the events that each group organizes constitute a chain, with the same amount of states, denoted as m, although users in some groups do not attend some of the events in their chain. Suppose that a user of the k^{th} group is attending the y-th event now, then the possible transitions will be to $y+1, y+2, ..., i$ and the corresponding probabilities will be $p(k)_{y(y+1)}$, $p(k)_{y(y+2)}$, ... ,$p(k)_{y(i)}$. These probabilities are depicted in the figure as $p(k)_{rs}$ where k represents the group and rs shows the transition from event r to event s.

5) Prediction Component: The goal of the Prediction Component, for a given group and feature, is to estimate the transition probabilities among the events for each group. Assume that the social network consists of k groups, as discussed in Section IV-A3. We exploit the MMM model for the individual groups since each group consists of different users (based on the features) and with different behavior. We make the assumption that users of the same groups would have a similar behavior. MMM has an unobservable parameter, which is the fixed group of the user. Specifically, the user's group defines the model that created the transition, namely if a user belongs to the k^{th} group, then we say that the k^{th} model caused the transition. The probability distribution sets a latent vector z for each transition. The latent vector z_n describes the n^{th} user's group and $z_{n,k} = 1$ when the k^{th} model caused the transition of n^{th} user, otherwise $z_{n,k} = 0$.

Computation of Probabilities. First, we define the probability of user n to attend a specific event:

$$P(w_{n,i}) = \sum_z P(w_{n,i}, z_n) \tag{2}$$

where $P(w_{n,i})$ represents the probability of user n being at a specific event, in state i. Also from the product rule we extract the joint probability of z_n and $w_{n,i}$ as:

$$P(z_n, w_{n,i}) = P(w_{n,i}|z_n)P(z_n) \tag{3}$$

The joint probability $P(z_n, w_{n,i})$, shows the probability of the n^{th} user being at a particular event and belonging to a particular group and due to symmetric rules: $P(z_n, w_{n,i}) = P(w_{n,i}, z_n)$. Thus, using equations 2 and 3 we take that:

$$P(w_{n,i}) = \sum_z P(w_{n,i}|z_n)P(z_n) \tag{4}$$

From equation 4 we infer that we should find the probabilities $P(z_n)$ and $P(w_{n,i}|z_n)$ to compute $P(w_{n,i})$. The marginal distribution over z_n ($P(z_n)$) is specified in terms of the mixing coefficient π_k, such that $P(z_{n,k} = 1) = \pi_k$ where the parameters $\{\pi_k\}$ must satisfy $0 \leq \pi_k \leq 1$ together with $\sum_{k=1}^{K} \pi_k = 1$ because the values of π_k are probabilities. Thus, the marginal distribution of z_n is defined as:

$$P(z_n) = \prod_{k=1}^{K} \pi_k^{z_{n,k}} \tag{5}$$

where $P(z_n)$ shows the probability the n^{th} user to belong to a specific group.

Similarly, we compute the probability $P(w_{n,i}|z_n)$, that represents the probability of user n being at a specific event given that he belongs to a particular group. We define the transition probability $p(k)_{rs}$ from event r to s under the k group, where $\sum_s p(k)_{rs} = 1$. Thus, the conditional distribution of $w_{n,i}$ given a particular value for z_n is:

$$P(w_{n,i}|z_n) = \prod_{k=1}^{K} (p(k)_{rs}^{w_{n,i-1,r}w_{n,i,s}})^{z_{n,k}} \tag{6}$$

The joint distribution $P(z_n, w_{n,i})$ is given by $P(w_{n,i}|z_n)P(z_n)$ and the marginal distribution of $w_{n,i}$ is then obtained by summing the joint distribution over all possible states of z_n:

$$P(w_{n,i}) = \sum_z P(w_{n,i}|z_n)P(z_n) = \sum_{k=1}^{K} \pi_k p(k)_{rs}^{w_{n,i-1,r}w_{n,i,s}} \tag{7}$$

In equation 7, π_k represents the prior probability of picking the k^{th} component and the probability $p(k)_{rs}^{w_{n,i-1,r}w_{n,i,s}}$ represents the probability of $w_{n,i}$ conditioned on $z_{n,k} = 1$ as can be observed in equation 6.

We also need to define the posterior probabilities $P(z_{n,k} = 1|w_{n,i})$, which represent the probability the n^{th} user belonging to group k on condition that he is at a specific event. These probabilities, also known as responsibilities, are computed by the Bayes theorem:

$$\gamma_k(w_{n,i}) = P(k|w_{n,i}) = \frac{\pi_k \prod_{i,rs} p(k)_{rs}^{w_{n,i-1,r}w_{n,i,s}}}{\sum_{k'} \prod_{i,rs} p(k')_{rs}^{w_{n,i-1,r}w_{n,i,s}}} \tag{8}$$

where $k = z_{n,k} = 1$ since, as we referred above, $z_{n,k} = 1$ when the k^{th} model causes the transition of user n.

Following that, we should estimate parameters π_k, $p(k)_{rs}$. One common criterion for determining the parameters in a probability distribution using an observed dataset is to find the parameter values that maximize the likelihood function. In our model the likelihood function is the probability $P(w_{n,i})$. In practice, it is more convenient to maximize the log of the likelihood function, since the logarithm is a monotonically increasing function, and so maximizing the log of a function is equivalent to maximizing the function itself. Consequently, logarithmic likelihood L is given as:

$$L = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \prod_{i,rs} p(k)_{rs}^{w_{n,i-1,r}w_{n,i,s}} \tag{9}$$

As we referred above, parameters π_k and $p(k)_{rs}$ will satisfy the following constrains: $\sum_k \pi_k = 1$ and $\sum_s p(k)_{rs} = 1$.

Maximizing L using Lagrange multipliers. In order to find the maximum of the function L, subject to equality constraints, we exploit the method of Lagrange multipliers[19] which is used to solve maximization problems. We introduce two new variables λ and μ called Lagrange multipliers and study the Lagrange function defined as:

$$\Lambda = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \prod_{i,rs} p(k)_{rs}^{x_{n,i-1,r}x_{n,i,s}} + \lambda\big(\sum_{k} \pi_k - 1\big) + \mu\big(\sum_{s} p(k)_{rs} - 1\big) \tag{10}$$

and we solve the following:

$$\nabla_{\pi_k, p(k)_{rs}, \lambda, \mu} \Lambda(\pi_k, p(k)_{rs}, \lambda, \mu) = 0 \tag{11}$$

Setting the derivative of Λ with respect to π_k to zero and using the constraint $\sum_k \pi_k$=1, we eliminate λ and obtain:

$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} \frac{\pi_k \prod_{i,rs} p(k)_{rs}^{w_{n,i-1,r}w_{n,i,s}}}{\sum_{k'} \prod_{i,rs} p(k')_{rs}^{w_{n,i-1,r}w_{n,i,s}}} = \frac{1}{N} \sum_{n=1}^{N} \gamma_k(w_{n,i}) \tag{12}$$

So the mixing coefficient π_k for the k^{th} group is given by the average responsibility that the group takes for explaining the user's attendances in organized events.

Setting the derivative of Λ with respect to $p(k)_{rs}$ to zero and taking into account that $\sum_s p(k)_{rs} = 1$, we eliminate the parameter μ and obtain the transition probability $p(k)_{rs}$:

$$p(k)_{rs} = \frac{\sum_{n=1}^{N} w_{n,i-1,r} w_{n,i,s} \gamma_k(w_{n,i})}{\sum_{n,s} w_{n,i-1,r} w_{n,i,s} \gamma_k(w_{n,i})} \tag{13}$$

Setting Variables using EM. Apart from the parameters, we should estimate the latent variable of MMM simultaneously. This is achieved by using the Expectation-Maximization(EM) algorithm which is an iterative method for finding maximum likelihood estimates of parameters in statistical models, where the model depends on unobserved latent variables. Thus, EM is used for determining the variables and computing the transition probabilities within the groups.

The algorithm involves two steps. The E-step is useful for calculating the expectation values of z ($\gamma_k(w_{n,i})$) using the observations and the current estimates of the values for the parameters. The M-step updates the parameters by maximizing the expectation found on the E-step. Thus, we first compute the latent variable $\gamma_k(w_{n,i})$ in the E-step using equation 8. In the M-step, we update the parameters π_k, $p(k)_{rs}$, using equations 12,13. Then, we check for convergence of either the parameters or the log likelihood $\log P(X|\pi, p(k)_{rs})$. If the convergence criterion is not satisfied, we return to the E-step and continue with the iterations. The log-likelihood is computed as follows:

$$\log P(X|\pi, p(k)_{rs}) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \prod_{i,rs} p(k)_{rs}^{w_{n,i-1,r}w_{n,i,s}} \tag{14}$$

which is similar to equation 9 but each time we update the new values of π_k and $p(k)_{rs}$. Our algorithm is summarized in Algorithm 1.

Finally, whenever the system is updated by inserting additional events (states) or user attends, the algorithm can be triggered again with initial values the previously computed π_k, $p(k)_{rs}$, to re-estimate the parameters.

Algorithm 1 Prediction of Next Event Attencdance

Input: group k, state i
Initialize the parameters π_k and $p(k)_{rs}$ randomly
repeat
  (E-Step) Compute the responsibilities $\gamma_k(w_{n,i})$
  (M-Step) Re-estimate the parameters π_k, $p(k)_{rs}$
until ($\log P(X|\pi, p(k)_{rs})$ OR $p(k)_{rs}$ converges)

B. Suggestion Phase

The suggestion phase is triggered either on demand by the user, or periodically to update the Next Event Suggestion at the user device, as described below. Additionally, a user can provide feedback whether he plans to visit the suggested event, which is forwarded to our middleware using the `insert()` function, as a JSON formatted object. This information is then pushed asynchronously (using the post() function) by PRESENT to the social networks that the user belongs to, through their APIs (section IV-A2).

1) Next Event Suggestion Component: Each user that aims to exploit our middleware should install the PRESENT Mobile Application to receive personalized event suggestions. The goal of the application is to predict and suggest the next event for a user to attend through a simple widget (figure 2).

When the user initiates the application for the first time he needs to select and connect with all the social networks that he belongs to, through the graphical interface. Whenever the application needs to provide such information to the user, it executes the function: `retrieve (group, feature, time)`. Hence, it requests the subset of the chain for the `group` that the user belongs to and for a specific `feature` (selected by the user), that includes all events after `time`. This request is performed by sending a JSON request object from the device to a well known address, binded by the Prediction Component of the respective PRESENT server. For instance, a user that belongs to the second group in figure 3, can execute the retrieve function, to retrieve the subset of the chain of his group, depending on the selected feature and time, which is illustrated with the blue cloud. The information requested is returned to the mobile device through a JSON object. That way, we do not publish private user information about the attends since we do not share explicit attendance information.

Afterwards, the Next Event Suggestion Component is triggered on the mobile device to predict the following event for a user to attend based on the probabilities $p(k)_{rs}$, for each transition r,s under the evaluated group k. The result of the prediction is defined as the transition $r \rightarrow s, \forall s$ with the highest probability $p(k)_{rs}$ where r is the last event that user n has visited. Thus, event s will appear at the user's device.

Finally, the user can provide feedback to the PRESENT system, whether he/she plans to attend the predicted event. The feedback is provided using the `insert(user, event)` function that encapsulates it to a JSON object. The JSON object is sent to the User Profiles Component to update the user's set of attended events, and trigger the `post` function to push this information to the social networks. The social networks we are interested in, support the HTTP protocol to push the data and thus, we need to produce POST requests using JSON structured data, to insert data to these networks.

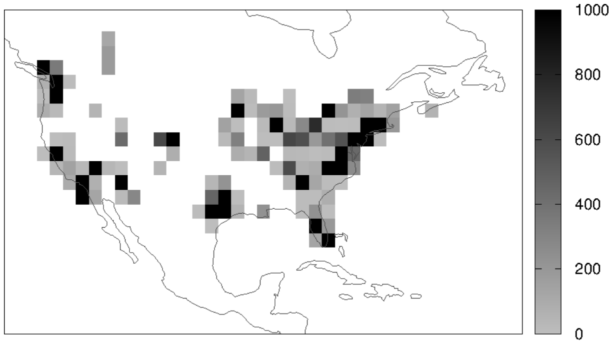

Fig. 4. HeatMap of the Meetup events in USA.

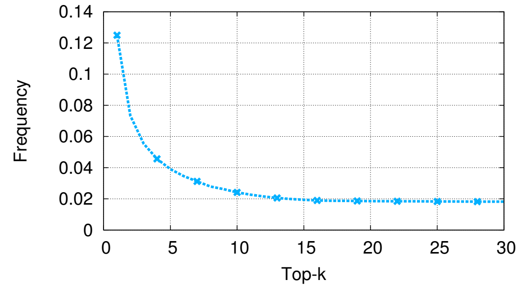

Fig. 5. Frequency of Top-K locations

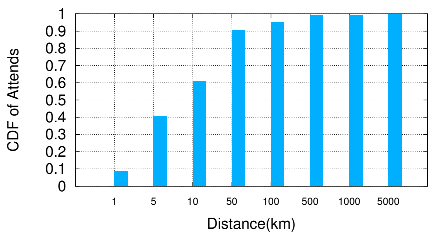

Fig. 6. CDF of the Distance among Users and Attends

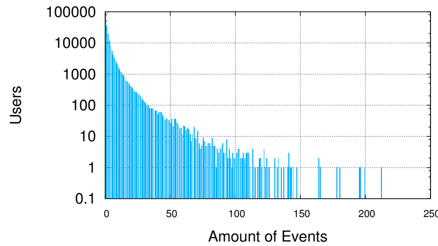

Fig. 7. Amount of Event attendances for the Users

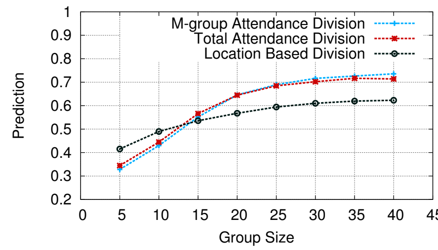

Fig. 8. Prediction over different Group Sizes

C. Discussion

In this paper we focus on understanding the features that shape human crowd behavior at social events, to quantify the similarity of the user behavior with other members of the social groups, and to make personalized recommendations. Focusing on the scalability aspect would be important if a lot of responses were generated in real-time for the events, and we would need to update the parameters constantly. During our experiments, with a real dataset that contains social data of more than a year, we did not face such problems. However, we could employ different engines for each geographical region or evaluate users under different levels of granularity (e.g., only high-attendance groups), to handle such issues.

V. Experimental Evaluation

A. Empirical Study

In this study, we consider a real-world dataset obtained from the popular Meetup social network. The dataset contains the history of all events that took place from 13 March 2010 until 31 July 2011. It includes 205684 users, which belong to 2578 active M-groups. Each user can be registered in many different M-groups. Each M-group corresponds to a topical category depending on the themes, from the categories in table I. In figure 4 we provide the distribution of the events in USA to better visualize social event activity across the cities. We focus on the USA since the vast majority of the events (77119 out of 89952) took place there. Note that locations with more than 1000 events, are assigned with the same color, to better illustrate the distribution of the locations.

As we show in the following sections, our analysis offers empirical evidence that the features selected for dividing the users into groups are correlated with specific user behaviors.

Understanding Human Crowd Behavior First, we perform a study on the Meetup social network to understand the behavior of the human crowd. This study enables us to determine the factors that differentiate user attendance behavior at events and explain the reasons that lead us to divide users into feature-based groups.

Location. In figure 5 we present the frequency of the top-K locations per user. We only present up to top-30 since the frequency of the following top-k locations is too small (less than 0.5%). As can be observed, many attends for each user are associated with a few places. For instance 12.5% attends in Meetup are located in one place, which means that users visit only a few places frequently but they visit many locations overall, as can be inferred by the slope at the tail of the figure.

Figure 6 presents the Cumulative Distribution Function (CDF) of the distance among the "neighborhood area of the users" and the locations they visit. The neighborhood area of each user represents the centroid of the locations he/she has visited. As can be observed Meetup users typically visit events close by since 60% of the users visit events within 10km and 95% of the users visit events within 100km.

Attendance. Figure 7 shows the amount of users and the corresponding number of attended events. As can be observed, the vast majority has attended fewer than 50 events. However, one user has attended more than 200 events. Thus, such a different behavior provides the insight that we need to evaluate such users in separate groups. Such a division also allow us to evaluate outliers separately.

B. Evaluation of PRESENT

In this section we study the performance of our prediction approach in the PRESENT system. The experimental evaluation focuses on the following parameters: (i) **Prediction Accuracy**, (ii) **Total States**, (iii) **Total Users**,(iv) **Execution times** and (v) **Accuracy levels per Category**. We compare PRESENT with two state-of-the-art techniques: (i) a technique that considers only individual features for the groups, proposed in [8], and we chose to compare it with the "Historical Venues" feature that was shown to perform better than any of the individual features, (ii) the collaborative filtering approach [11] which is a common approach for recommendation systems.

In our prediction approach we evaluate all transitions from an event that a user has attended to the next event that he attended. We accept as a correct prediction if the user sequence of event attendance is the one that we have predicted (*i.e.*, the transition from state a to state b with the highest probability among all transitions $(a, x)\forall x$), and as a false prediction otherwise. We do not consider users with fewer than two transitions and in order to qualify a group for the prediction, the group needs to have at least three users. Finally, we consider 66% of the users as the training set and the rest of the users constitute the evaluation group.

We evaluate the prediction accuracy by considering that users follow peers with similar behaviors within the group based on the attendance and location features. We divide the users into equal subsets based on their attendance to the events and based on the centroid of the location of the events attended. Producing groups based on user attendance is achieved through sorting. However, to extract groups based on the location we scan the spatial area and for each user we discover his k-Nearest-Neighbors [20]. We define these k users as a group G_j, remove them from the user set and continue scanning the area. The presented results are averages over 3 runs.

PRESENT Efficiency. In figure 8 we present the average prediction for the individual users for the top 500 M-groups, based on their attendance. The selection of 500 was based on the amount of users per group, to have enough users (*i.e.*, at least 135 users) for the prediction and the evaluation. The top 500 M-groups attendance ranges from 135 to 2100 users. We present three division strategies: (i) we divide the users into groups based on the amount of user attendances within the evaluated M-group (Attendance Criterion), (ii) we divide the users within a group based on their total attendances in the dataset (Attendance Criterion), and (iii) we divide the users based on the centroid of the locations where they attended events (Spatial Criterion). The figure shows that attendance-based strategies behave similar, with the first one having an advantage for smaller groups, with 1.6% better prediction when dividing users into groups of 5, and the second technique achieving a higher prediction for larger groups. Increasing the size of the groups increases the prediction, since more data become available for each group and the users are assigned to feature based groups with similar attendance frequency. The prediction starts to converge at 72% when the size of groups reaches 30. The third strategy shows a high advantage when dividing the users into smaller groups, since the users that attend events in similar venues are grouped together. However, when the amount of groups increases the prediction does not increase with the same slope as in the other techniques leading to lowest prediction accuracy. Since the first strategy performs better, we present this strategy in the following experiments.

Figure 9 illustrates the prediction accuracy for each of the individual top-500 M-groups and for different group sizes, sorted based on the accuracy. As can be observed, dividing the groups to 20 and 30 users highly increases the accuracy. However, it leads to some of the groups not being able to qualify for the evaluation, as shown in the tail of the lines, since we require at least $3 * group - size$ active users at each of these groups. Nevertheless, this evaluation shows that each group can be tuned individually to provide maximum prediction according to the amount of active members.

Experimental Parameters. Figure 10 illustrates the amount of states for each of the top-500 evaluated M-groups in sorted order. Hence, the states reach up to 201 and more than 350 M-groups include more than 50 states. Keep in mind that we only keep states where at least one user has attended. This amplifies the importance of our prediction accuracy since from each state the user can move to each one of the following states. Note that the probability of a haphazard prediction, is $\frac{1}{x-y}$, when the user resides in state y out of the total states x. Also, figure 11 presents the corresponding amount of users for the M-groups we consider in a descending order. Thus, only 46 out of the 500 M-groups have fewer than 100 users and the top 100 M-groups have more than 400 users. We note that figures 9, 10, 11 are individually sorted and they do not present the respective data for each M-group. Intuitively, a small amount of states and users does not lead to low predictions.

Execution Time. We present the execution time of the Model Generator component as the iterations and the states (events) increase, for the M-group with the highest attendance. As can be seen in figure 12, the execution time remains almost the same to the number of iterations, until the probabilities converge. We present the execution times only up to 10 iterations, since we have performed all of our experiments with 10 iterations, that were enough to define the probabilities. As the figure shows, the execution time depends on the state that we have reached, since it considers only the probabilities of the previous states. However, the execution time remains low (less than 120 milliseconds at all times), especially when considering that these states reflect a time interval of more than a year. The depicted spikes occur due to the small execution time along with the processes that run in parallel.

Prediction per Category. In figure 13 we present the prediction level at each category based on our approach. As can be seen, categories like Hobbies and Crafts (15) have increased accuracy since the members of a specific hobby attend most events (frequent members) or the most important gatherings (not-frequent members). On the other hand categories like Food and Drink (10) have lower prediction accuracy, as the users choose to attend a meeting based on subjective criteria. However, the prediction accuracy is high and especially when we divide the groups into 30 users.

Comparison. We compare PRESENT with two state-of-the-art techniques for recommendation systems in social network settings such as Foursquare and Facebook [12]: (i) collaborative filtering (CF) [11], where we predict that the next event for user n will be the same with user n' that achieves the highest score, based on the common attends $(score(n, n') = \frac{2*|A_n \cap A'_n|}{|A_n|+|A'_n|}$, where A_n denotes the set of events that user n has attended) and (ii) the strategy of predicting the user's next check-in based on the individual features, proposed in [8]. As shown in this paper, the optimal prediction was achieved with the historical venues feature, so we implemented that strategy. Thus, we predict the user's next check-in based on the venue he has visited mostly.

Figure 14, presents the comparison for the top-100 M-groups of each category. As can be observed PRESENT

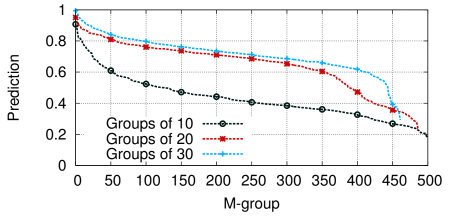

Fig. 9. Prediction for the Top-500 groups over different Group Sizes

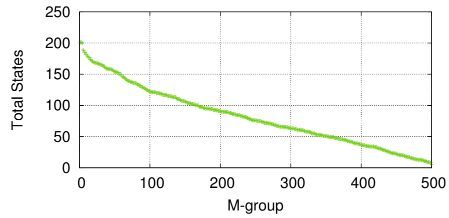

Fig. 10. States of the evaluated groups

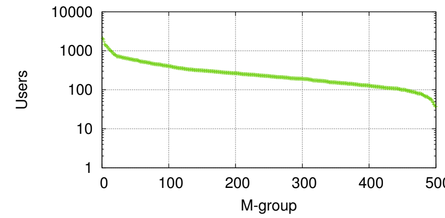

Fig. 11. Users of the evaluated groups

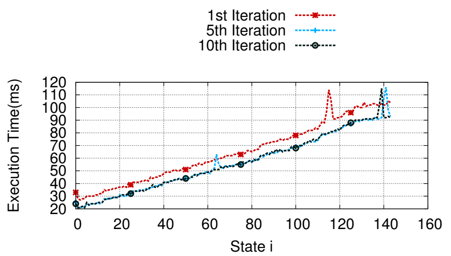

Fig. 12. Execution Times

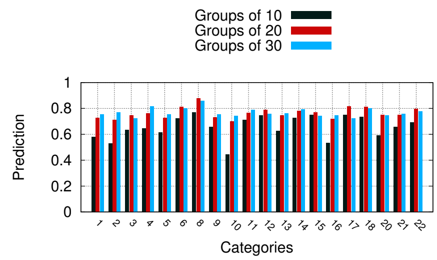

Fig. 13. Prediction Accuracy Per Category

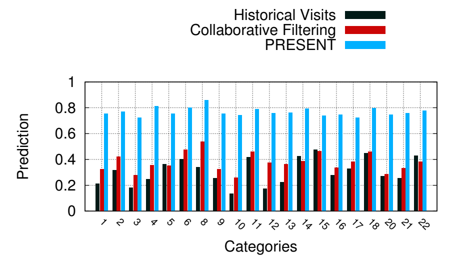

Fig. 14. Prediction Accuracy Comparison Per Category

outperforms both techniques in every individual category. The historical-visits prediction achieves up to 47.6% (15-th category) and the CF approach achieves a prediction up to 53.8% (8-th category), while PRESENT achieves a worst prediction of 72.5% and scales up to 85.8% at the 8-th category. This is mainly because CF depends on the events that the "closest user" will participate, while the Historical Visits approach depends on the venues that a user visits frequently. Thus, when users provide a few RSVPs, these techniques face a cold start, and they cannot make accurate predictions. These approaches work well when providing a list of recommendations but they are not as efficient for our problem, since recommendation systems suggest events based on rankings but the next event is not always the highest ranked event (users might visit another event previously). PRESENT depends on the behavior of a group of users, that enables us to make accurate predictions, since a cold start may occur when we instantiate the group rather than when a user joins the system.

VI. Related Work

Several event recommendation techniques that utilize metrics to decide for the events to be recommended to the user, exist in the literature [4], [5], [6]. However, they do not consider the behavior of social groups and evaluate each event individually, rather than recommending the next event for each user. Among the most common techniques for event recommendation is the collaborative filtering [11], [12]. However, as we show in our experiments these techniques cannot capture well the event attendance of a user, influenced by social interactions. Other event recommendation techniques consider users that exhibiting similar behaviors [21], combine collaborative filtering approaches with content-based algorithms [13] and recommendation systems for geographical locations [22], [23]. However, none of these systems considers the sequence of the recommended items (events). In [24] and [25] they propose approaches to recommend routes and trips that consider preferences from multiple users, but they do not aim to predict them. Authors in [10] exploit Meetup to examine the event recommendation based on the social graph and the RSVPs. PRESENT differs since we focus at predicting the following event attendance for each user. In [26] they formalize the trade-offs between accuracy and differential privacy of personalized social recommendations, but they consider individual recommendations rather than the next event to attend.

Several papers have been proposed in the context of predicting the human mobility [27], [28]. However, most of these works depend on the repetition of the users on visiting places and fail to predict irregular movements, such as the social events. Moreover, they do not consider the movement of other users along with existing social ties. Authors in [29], exploit the MMM model to predict the user's next move. However, their setting assumes a set of finite and predefined states where the user can go back, while the chains in our model change over time and the user can only move to a future event. Moreover, human mobility differs from the user event attendance, since a user that moves spatially needs to cross over specific places (states) to reach his target position. On the contrary, users can skip several events without affecting the following events that they will participate. Authors in [30] focus on the next location prediction for human trajectories by considering moving behaviors of users. However, as we explained, human mobility differs from event attendances.

Authors in [8] aim to predict the users next check-in, based on different features of the user's check-ins. Similarly, in [7] they exploit HMM to determine the user's category and location of his next check-in. However, our work differs, since we consider the group behavior rather than the user's individual characteristics. Moreover, in contrast to PRESENT such approaches face a cold start problem when the user has a few check-ins. In [31] they propose NextPlace, a prediction framework to forecast user behavior in different locations, but they focus on the temporal predictability of the users presence and they do not consider the transitions among locations. Liao *et al* in [9] aim to predict the events that a user will attend, based on user's event participation and the physical proximity with other users for offline ephemeral social networks. Our work differs since our model considers users as groups with similar features and consequent events, rather than evaluating individual short-term user behavior for parallel events.

Authors in [32] develop a prediction framework based on social data to predict the location of a user at a given time.

However, they do not consider the group's impact and they focus on user mobility for the next few hours. In [33] the authors attempt to infer the influence of the social environment to the user's movements, in terms of distance. In our paper we evaluate groups of users, according to different criteria, and we examine if these groups affect the event attendance by the users. In [34] they provide an in-depth analysis of geo-social influence in LBSNs. They assume that the behavior of a social group affects the users, but they do not focus on predicting the user behavior that derives from that effect.

In our previous work [35] we exploited GPS traces to identify real-world events efficiently. Our current work differs since we consider the events as given from our dataset and our goal is to suggest the following event for a user to attend considering the behavior of the group that he belongs to. Also, in [14] we proposed an approach to combine spatiotemporal data from heterogeneous types of networks. PRESENT differs since we present a middleware that aims to predict the next event that users will attend using multiple networks.

VII. Conclusions

In this paper we have presented PRESENT, our middleware that exploits the social behavior of the human crowd to identify group attendance behaviors and predict the next event for a user to attend. Our experimental study with a real-world dataset illustrates the contributions of our approach, and verify its practicality and efficiency. For our future work we plan to exploit whether explicit social interactions among the users can further contribute to improve our prediction accuracy.

Acknowledgment

This research has been co-financed by the European Union (European Social Fund ESF) and Greek national funds through the Operational Program Education and Lifelong Learning of the National Strategic Reference Framework (NSRF) - Research Funding Program: Thalis-DISFER, Aristeia-MMD Investing in knowledge society through the European Social Fund, the FP7 INSIGHT project and the ERC IDEAS NGHCS project.

References

[1] "Foursquare," https://foursquare.com/.

[2] "Meetup," http://www.meetup.com/.

[3] "Twitter," http://www.twitter.com/.

[4] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen, "Lcars: a location-content-aware recommender system," in *KDD*, Chicago, IL, USA, August 2013, pp. 221–229.

[5] D. Quercia, N. Lathia, F. Calabrese, G. Di Lorenzo, and J. Crowcroft, "Recommending social events from mobile phone location data," in *ICDM*, Sydney, Australia, December 2010, pp. 971–976.

[6] T. De Pessemier, J. Minnaert, K. Vanhecke, S. Dooms, and L. Martens, "Social recommendations for events," in *RSWeb*, Hong Kong, Oct 2013.

[7] J. Ye, Z. Zhu, and H. Cheng, "Whats your next move: User activity prediction in location-based social networks," in *SDM*, Austin, Texas, USA, May 2013, pp. 171–179.

[8] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo, "Mining user mobility features for next place prediction in location-based services," in *ICDM*, Brussels, Belgium, December 2012, pp. 1038–1043.

[9] G. Liao, Y. Zhao, S. Xie, and P. S. Yu, "An effective latent networks fusion based model for event recommendation in offline ephemeral social networks," in *CIKM*, San Francisco, CA, Oct 2013, pp. 1655–1660.

[10] 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 *SIGKDD*, Beijing, China, August 2012, pp. 1032–1040.

[11] E. Minkov, B. Charrow, J. Ledlie, S. Teller, and T. Jaakkola, "Collaborative future event recommendation," in *CIKM*, Canada, October 2010, pp. 819–828.

[12] M. Kayaalp, T. Ozyer, and S. Ozyer, "A collaborative and content based event recommendation system integrated with data collection scrapers and services at a social networking site," in *ASONAM*, Athens, Greece, July 2009, pp. 113–118.

[13] C. Cornelis, X. Guo, J. Lu, and G. Zhang, "A fuzzy relational approach to event recommendation." in *IICAI*, December 2005.

[14] S. Karanikolaou, I. Boutsis, and V. Kalogeraki, "Poster: Understanding event attendance through analysis of human crowd behavior in social networks," in *DEBS*, Mumbai, India, May 2014, pp. 322–325.

[15] X. Liu, Y. Tian, M. Ye, and W.-C. Lee, "Exploring personal impact for group recommendation," in *CIKM*, Maui, HI, Oct 2012, pp. 674–683.

[16] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, 2006, vol. 1.

[17] H. Mannila, H. Toivonen, and A. I. Verkamo, "Discovery of frequent episodes in event sequences," *Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 259–289, 1997.

[18] J. Shi, N. Mamoulis, D. Wu, and D. W. Cheung, "Density-based place clustering in geo-social networks," in *SIGMOD*, Snowbird, UT, USA, June 2014, pp. 99–110.

[19] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1999.

[20] S. A. Nene and S. K. Nayar, "A simple algorithm for nearest neighbor search in high dimensions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 9, pp. 989–1003, Sep. 1997.

[21] M. Ye, X. Liu, and W.-C. Lee, "Exploring social influence for recommendation: a generative model approach," in *SIGIR*, Portland, OR, August 2012, pp. 671–680.

[22] J.-D. Zhang and C.-Y. Chow, "Core: Exploiting the personalized influence of two-dimensional geographic coordinates for location recommendations," *Information Sciences*, vol. 293, pp. 163–181, 2015.

[23] J. Bao, Y. Zheng, and M. F. Mokbel, "Location-based and preference-aware recommendation using sparse geo-social networking data," in *SIGSPATIAL*, Redondo Beach, CA, November 2012, pp. 199–208.

[24] L. Guo, J. Shao, K. L. Tan, and Y. Yang, "Wheretogo: Personalized travel recommendation for individuals and groups," in *MDM*, vol. 1, Brisbane, Australia, July 2014, pp. 49–58.

[25] E. H.-C. Lu, C.-Y. Chen, and V. S. Tseng, "Personalized trip recommendation with multiple constraints by mining user check-in behaviors," in *SIGSPATIAL*, Redondo Beach, CA, November 2012, pp. 209–218.

[26] A. Machanavajjhala, A. Korolova, and A. D. Sarma, "Personalized social recommendations: accurate or private," *Proceedings of the VLDB Endowment*, vol. 4, no. 7, pp. 440–450, 2011.

[27] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti, "Wherenext: a location predictor on trajectory pattern mining," in *SIGKDD*, Paris, France, June 2009, pp. 637–646.

[28] P.-R. Lei, T.-J. Shen, W.-C. Peng, and J. Su, "Exploring spatial-temporal trajectory model for location prediction," in *MDM*, vol. 1, Lulea, Sweden, June 2011, pp. 58–67.

[29] A. Asahara, K. Maruyama, A. Sato, and K. Seto, "Pedestrian-movement prediction based on mixed markov-chain model," in *SIGSPATIAL*, Chicago, IL, USA, November 2011, pp. 25–33.

[30] J. J.-C. Ying, W.-C. Lee, and V. S. Tseng, "Mining geographic-temporal-semantic patterns in trajectories for location prediction," *TIST*, vol. 5, no. 1, pp. 2:1–2:33, 2014.

[31] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell, "Nextplace: a spatio-temporal prediction framework for pervasive systems," in *Pervasive Computing*, San Francisco, CA, June 2011, pp. 152–169.

[32] M.-F. Chiang, W.-Y. Zhu, W.-C. Peng, and P. S. Yu, "Distant-time location prediction in low-sampling-rate trajectories," in *MDM*, Milan, Italy, June 2013, pp. 117–126.

[33] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *SIGKDD*, San Diego, CA, USA, August 2011, pp. 1082–1090.

[34] C. Zhang, L. Shou, K. Chen, G. Chen, and Y. Bei, "Evaluating geo-social influence in location-based social networks," in *CIKM*, Maui, HI, October 2012, pp. 1442–1451.

[35] I. Boutsis, V. Kalogeraki, and D. Gunopulos, "Efficient event detection by exploiting crowds," in *DEBS*, Arlington, Texas, USA, June 2013, pp. 123–134.