Activity Recommendation with Partners

WENTING TU, Shanghai University of Finance and Economics
DAVID W. CHEUNG and NIKOS MAMOULIS, The University of Hong Kong
MIN YANG, Chinese Academy of Sciences
ZIYU LU, The University of Hong Kong

Recommending *social activities*, such as watching movies or having dinner, is a common function found in social networks or e-commerce sites. Besides certain websites which manage activity-related locations (e.g., foursquare.com), many items on product sale platforms (e.g., groupon.com) can naturally be mapped to social activities. For example, movie tickets can be thought of as *activity items*, which can be mapped as a social activity of "watch a movie." Traditional recommender systems estimate the degree of interest for a target user on candidate items (or activities), and accordingly, recommend the top-k activity items to the user. However, these systems ignore an important social characteristic of recommended activities: people usually tend to participate in those activities with friends. This article considers this fact for improving the effectiveness of recommendation in two directions. First, we study the problem of *activity-partner recommendation*; i.e., for each recommended activity item, find a suitable partner for the user. This (i) saves the user's time for finding activity partners, (ii) increases the likelihood that the activity item will be selected by the user, and (iii) improves the effectiveness of recommender systems to users overall and enkindles their social enthusiasm. Our partner recommender is built upon the users' historical attendance preferences, their social context, and geographic information. Moreover, we explore how to leverage the partner recommendation to help improve the effectiveness of recommending activities to users. Assuming that users tend to select the activities for which they can find suitable partners, we propose a *partner-aware activity recommendation* model, which integrates this hypothesis into conventional recommendation approaches. Finally, the recommended items not only match users' interests, but also have high chances to be selected by the users, because the users can find suitable partners to attend the corresponding activities together. We conduct experiments on real data to evaluate the effectiveness of activity-partner recommendation and partner-aware activity recommendation. The results verify that (i) suggesting partners greatly improves the likelihood that a recommended activity item is to be selected by the target user and (ii) considering the existence of suitable partners in the ranking of recommended items improves the accuracy of recommendation significantly.

CCS Concepts: • **Information systems** → **Social recommendation**;

Additional Key Words and Phrases: Recommendation system, Location-based social network

Authors' addresses: W. Tu, School of Information Management and Engineering, Shanghai University of Finance and Economics, 777 Guoding Road, Shanghai, China; email: tu.wenting@mail.shufe.edu.cn; D. W. Cheung, N. Mamoulis, and Z. Lu, Computer Science Department, The University of Hong Kong, Pokfulam, Hong Kong; emails: {dchueng, nikos, zylu}@cs.hku.hk; M. Yang, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Xueyuan Avenue 1068, Shenzhen, China; email: min.yang1129@gmail.com.

https://doi.org/10.1145/3121407

ACM Reference format:
Wenting Tu, David W. Cheung, Nikos Mamoulis, Min Yang, and Ziyu Lu. 2017. Activity Recommendation with Partners. *ACM Trans. Web* 12, 1, Article 4 (September 2017), 29 pages.
https://doi.org/10.1145/3121407

1 INTRODUCTION

In real-world recommendation applications, many items are related to activities that people like to participate in with their folks. For example, items such as movie tickets and dinner discounts are related to social activities (watching movies and dining). We call such items (social) *activity items*. Activity items are commonly found in real-world e-commerce websites such as Groupon (www.groupon.com) and Meituan (www.meituan.com), as shown in the examples of Figure 1(1). Previous work on recommending activity items typically focused on utilizing past attendance behaviors (Ye et al. 2013; Noulas et al. 2011; Zheng and Xie 2010; Yin et al. 2015), social links between users (Scellato et al. 2011; Ye et al. 2010, 2012), and geographic information (Li et al. 2008; Levandoski et al. 2012; Liu et al. 2013) to predict the interests of users. Our work is the first to consider a special characteristic of social activities: people typically do not like to attend them alone. Indeed, more often than not, when considering attending a social activity, people seek partners to join them. Based on this, we extend the functionality of recommender systems in two directions that improve their effectiveness in suggesting activity items.

First, assuming that a recommendation service (e.g., Groupon) promotes a set of activity items to a user, we study the problem of also recommending suitable *activity partners* for the items. As Figure 1 shows, our suggestion is to combine an activity-item promotion platform with a social network platform to find activity partners for the items which can increase the likelihood that the recommended items will be selected by the users. The rationale is that, for items that people like to participate in with their folks, if the system recommends the items alone, the user may give up attending the activity (i.e., taking up the item) if s/he cannot immediately think of someone to invite to attend the activity together. Figure 1(3) illustrates the effectiveness of recommending activity partners via an example. Consider activity item "tickets of Bruno Mars' concert," for which the corresponding activity is "watching Bruno Mars' concert." Imagine that you have some interest in Bruno Mars' show; however, when you see the recommendation message, you have difficulty thinking who could be suitable partners for watching the show together. This could be a good reason for you to give up attending this activity since you do not feel like watching a concert alone. On the other hand, if the recommendation also includes suggestions for possible partners, you can try inviting them and enjoy the show together. In order to evaluate our hypothesis that users prefer to take activity items if they have partners to join them, we designed a simple questionnaire to collect feedback from real Web users. The results (shown in Section 4.1) demonstrate that the great majority of Web users would favor such an approach as opposed to a simple activity item recommender. In summary, we assert that including partner recommendations not only improves the quality of recommender systems, but may also increase the positive response rate of users, therefore improving the revenue of the involved businesses. To the best of our knowledge, so far there do not exist any previous studies or applications that include this important function. This motivates us to investigate methods for activity-partner recommendation. We first explore how historical attendance preferences, social context, and geographic information can be used to recommend activity partners. Then, we propose a method that analyzes historical records of users' preferences on activity partners to predict activity partners for new recommended activities. This is a reasonable methodology, since the past users' preferences on activity partners would be available after setting up an activity-partner recommendation system.

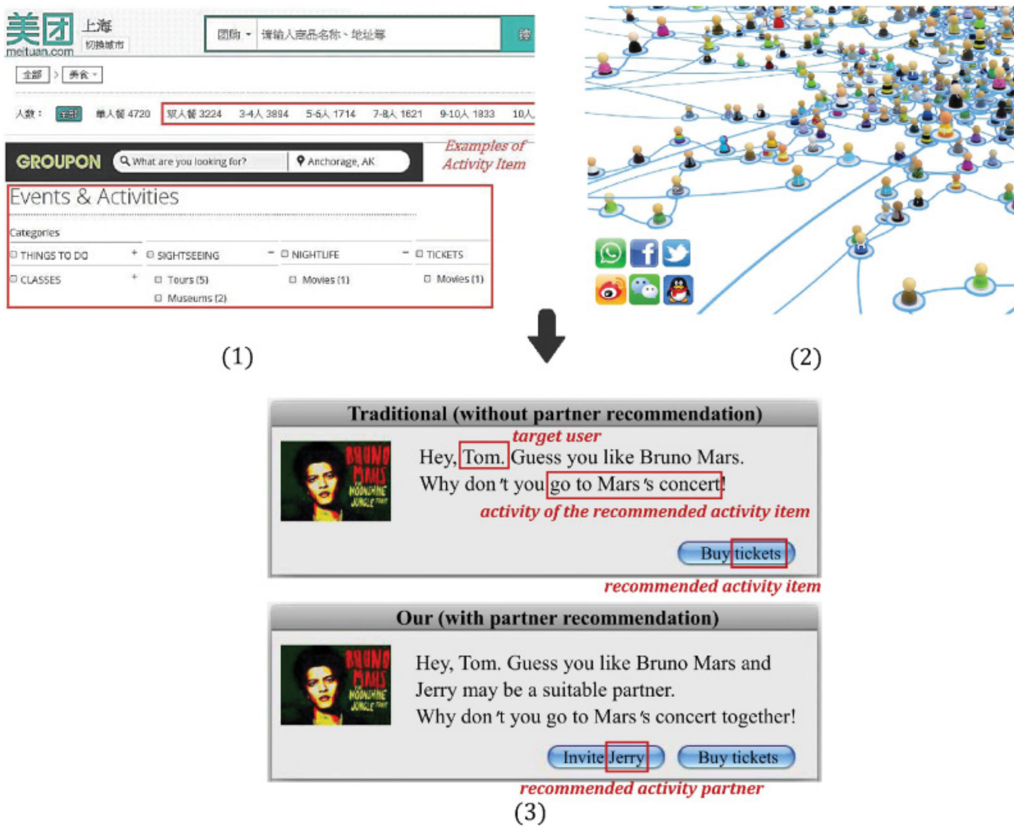


Fig. 1. Example of activity-partner recommendation.

Besides investigating how to recommend activity partners for any given item, our second direction of study in this article is how to utilize the fact that users like to attend activities with partners, in order to improve the quality of activity item recommendation. Our study is based on the assumption that users will prefer activities for which they can find suitable partners over activities for which they cannot find good partners. For example, consider a recommender system which suggests a list of activity items to a user that are expected to be the ones that she would prefer, based on the user's historical data. The user may, however, prefer items which she likes less than those in this list, but for which she could find partners. Based on this observation, we propose a *partner-aware activity recommendation* framework. Our framework first estimates the probability that a user can find partners for an activity and then uses this *partner probability* to adjust the recommendation order of activities. We expect that the list of recommended items (together with the corresponding partners) suggested by our framework is more useful to users, due to the fact that users not only are interested in the recommended activities, but also in whether they can find partners to attend them together.

Our experiments first include the results of a questionnaire which evaluates the needs of users for activity-partner recommendation. Then, we evaluate our proposed methods for partner recommendation and for partner-aware activity recommendation. We use datasets from location-based social networks to simulate a social-activity recommendation scenario. We select out the locations that can be mapped to social activities. The ground truth of activity partners can be extracted from users' check-in records and social links between users (refer to Section 4.2.3 for details). Using the ground-truth, we evaluate all proposed strategies for recommending activity partners. After evaluating partner recommendation, we study the effectiveness of partner-aware activity recommendation. Not only do we verify the assumption in our partner-aware framework, we also show that our framework improves conventional recommenders for activity promotion.

In summary, the contributions of this article are as follows:

— We bring in the idea of recommending suitable activity partners to users for social activities (or related products). The results of a survey which we have conducted confirms that real users appreciate the recommendation of activity partners together with the proposed items. We formulate the problem of activity-partner recommendation, accordingly.
— We study how to derive activity-partner recommendations based on the users' historical attendance behaviors, the social context of users, and geographic information of activities. Since such data are commonly tracked in current recommendation systems, our results can be used to set up an activity-partner recommender easily.
— We also propose a methodology for recommending activity partners based on past partner knowledge of users. In this direction, we extend conventional collaborative filtering techniques to make them more suitable for our problem.
— We adapt activity recommendation to consider not only the preference from users to activities, but also whether the users can find suitable partners to attend the recommended activities together. This results in a novel partner-aware activity recommendation model.
— We conduct an experimental evaluation based on real data that evaluates all proposed methods in terms of their ability to recommend suitable activity partners and improve recommendation effectiveness.

This article is a substantial extension of Tu et al. (2015), which proposes an activity-partner recommender for social activities. In this article, besides improving the activity-partner recommendation system, we also explore the impact of finding suitable partners to improve the effectiveness of recommending activities to users. More specifically, our new material compared to Tu et al. (2015) includes a new partner-aware activity recommendation model (Section 3), which assumes that users tend to select the activities for which they can find suitable partners. The model is evaluated in Section 4.4. We also add a new GCAPR strategy that utilizes geographic information for achieving activity-partner recommendation in Section 2. GCAPR is included in our performance evaluation (Section 4.3.2). Finally, we have added additional conventional activity recommenders for evaluating attendance preferences (Section 4.2.5) while we only used user-based Collaborative Filtering (CF) in Tu et al. (2015).

The remainder of this article is organized as follows. Section 2 describes our methods for activity-partner recommendation. Section 3 shows how we utilize the partner factor to improve conventional recommender systems for social activities. Section 4 includes our experiments. Section 5 presents related work. Finally, Section 6 concludes with a discussion about future work.

2 ACTIVITY-PARTNER RECOMMENDATION

In this section, we investigate the problem of recommending activity partners to a target user u_t to attend a given social activity item a_l together. Here, by *social activity items* we denote any real-world items related to *activities* people like to attend with their friends (e.g., watching an event, having dinner). We propose several solutions to recommend activity partners, based on different hypotheses. In Section 4.3, we will compare their performance on real-world datasets. As discussed in the Introduction, our motivation is to increase the probability that u_t will select a_l, assuming that a_l has been recommended to u_t. In other words, without partner recommendations,

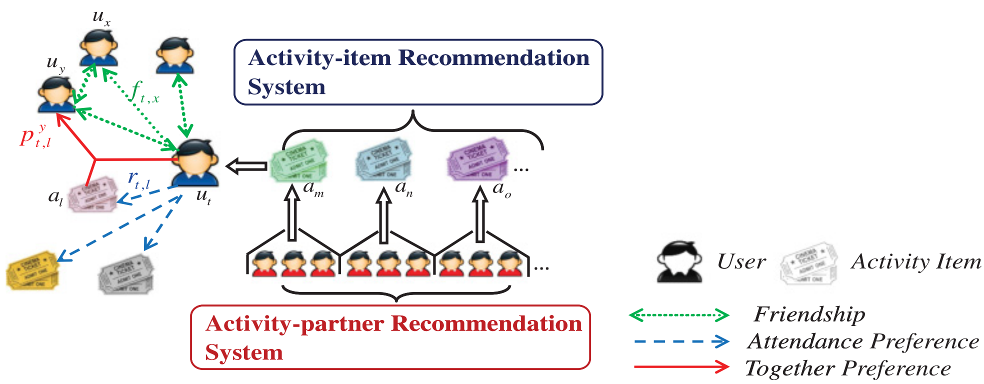


Fig. 2. Graphical explanation of activity-partner recommendation.

we assert that u_t has a higher probability to reject a_l (as indicated by our experiment in Section 4.1). After formally defining the activity-partner recommendation problem (Section 2.1), in Section 2.2 we show how we can consider various factors (attendance behavior, social context, geographic information) in defining the suitability of other users to be partners for u_t in attending a_l together. Finally, in Section 2.3 we discuss how historical information about the preferences of users in participating in activities together can be used toward finding suitable partners.

2.1 Problem Formulation

As illustrated in Figure 2, typically there are two types of objects (i.e., *users* and *items*) in recommendation systems. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_{n_u}\}$ be a set of users and $\mathcal{A} = \{a_1, a_2, \ldots, a_{n_a}\}$ be a set of activity items that can be recommended to users. Two common types of relationships exist among these entities. First, users have preferences for items, which are either explicit (e.g., users may *rate* some of the items) or implicit (preferences are estimated/predicted by a classic recommender system). Since, in our case, items are related to activities, we call the preference of users to items *attendance preference*. For each user u_t and activity item a_l, we denote by $af(u_t \rightarrow a_l)$ (abbreviated as $r_{t,l}$ in Figure 2) how much u_t prefers a_l. $af(u_t \rightarrow a_l)$ can take value from a range of integers (e.g., 1–5) or can be a binary number (i.e., $af(u_t \rightarrow a_l) = 1$ means that u_t likes a_l). Second, users can be connected to each other in a social network; we use $f_{i,j}$ to represent the *friendship* status between users u_i and u_j, i.e., $f_{i,j} = 1$ if u_i and u_j are friends and $f_{i,j} = 0$ otherwise.

Besides the above two types of relationships (i.e., friendship and attendance preference), we bring in another relationship, called ***together preference***, which indicates whether or how much a user prefers to attend a given activity item together with another user. For example, if Tom clicks the “Invite Jerry” button in the exemplary user interface in Figure 1(3), this indicates that Tom prefers to attend activity “Bruno Mars’ show” together with Jerry. We use $pf([u_t, a_l] \rightarrow u_x)$ (abbreviated as $p^x_{t,l}$ in Figure 2) to indicate how much user u_t prefers to attend the activity of a_l together with u_x. $pf([u_t, a_l] \rightarrow u_x)$ can take numerical or binary values, similar to the attendance preference defined above. For example, we can set the binary value of pf([Tom, tickets of Bruno Mars’ concert]→ Jerry) to 1 if Tom clicks the “Invite Jerry” button and to 0 if Tom does not click the button.

As Figure 2 illustrates, the objective of our work is as follows. For each activity item a_l recommended to a user u_t by a conventional recommender, predict the users’ together preference to all partner candidates on the activity item. In order to compute $pf([u_t, a_l] \rightarrow u_x)$, for any candidate partner u_x, we use any known friendship, attendance preference, and together preference relationships. Finally, we recommend to u_t the top-k partner candidates with the highest together preferences.

2.2 Utilizing Attendance Behavior, Social Context, and Geographic Information

The together preference $pf([u_t, a_l] \to u_x)$ relates a two-dimensional object $[u_t, a_l]$ to a user u_x. Although there is a large amount of work analyzing relationships between users and relationships between users and items, there is no previous work in recommender systems exploring this type of relationship.

A typical promotion platform of social activities keeps a record of three kinds of data: users' historical attendance records (hits), friendship between users, and the geographic information about the promoted activities. In this section, we will discuss how to utilize these data to implement partner recommendation, since they commonly exist in activity-recommendation platforms (Ye et al. 2013; Noulas et al. 2011; Zheng et al. 2009). This way, an activity-partner recommendation system can be trained in the case where we have not collected enough past together data from users.

2.2.1 Social Closeness Partner Hypothesis. The majority of web services nowadays allow users to establish friendship relationships between them. Thus, the most intuitive relationship between users is their social closeness. Here we use the *neighborhood overlap* (Adamic and Adar 2003) (commonly used owing to its low computational complexity) to model the social closeness $SC(u_t, u_x)$ between two users u_t and u_x. We argue that this user-user relationship is one factor that may help to predict together preference (e.g., $pf([u_t, a_l] \to u_x)$). The main assumption is that people prefer to attend activities with users who are socially close to them. Therefore,

$$pf([u_t, a_l] \to u_x) \propto SC(u_t, u_x) = \frac{\mathcal{F}^t \bigcap \mathcal{F}^x}{\mathcal{F}^t \bigcup \mathcal{F}^x}, \tag{1}$$

where $\mathcal{F}^t$ ($\mathcal{F}^x$) is the friends set of u_t (u_x). In order to recommend activity partners to a target user u_t, we can rank the activity-partner candidates u_x according to their social closeness to u_t and return the top ones as the recommended partners. We call this method *Social-Closeness based Activity-Partner Recommendation (SCAPR).*

2.2.2 Similar Interests Partner Hypothesis. The similarity between the interests/preferences of users (a.k.a. *user homophily*) is another important factor employed in classic recommender systems (Anderson et al. 2012). For recommending activity partners based on user homophily, we can rank the activity-partner candidates according to their similarity to the target user. This approach assumes that users prefer to participate in activities with people who have similar interests with them. For example, we can measure the cosine similarity between user-profile vectors. We call this method *Similar Interests based Activity-Partner Recommendation (SIAPR)*:

$$pf([u_t, a_l] \to u_x) \propto SI(u_t, u_x) = \cos(\overline{r_t}, \overline{r_x}) = \frac{\overline{r_t} \cdot \overline{r_x}}{||\overline{r_t}||||\overline{r_x}||}, \tag{2}$$

where vectors r_t and r_x capture the interests (set of preferred items) of u_t and u_x, respectively.

2.2.3 Also-Like Partner Hypothesis. Besides the above hypotheses, assuming that users prefer to attend an activity together with users who also prefer to attend the activity, we can rank the activity-partner candidates by their attendance preference to the activity item:

$$pf([u_t, a_l] \to u_x) \propto af(u_x \to a_l). \tag{3}$$

We call this method *Also-Like based Activity-Partner Recommendation (ALAPR).* The attendance preference of the activity-partner candidates to the activity item can be estimated by any activity-item recommendation system. For example, we can use user-based collaborative filtering (Schafer et al. 2007) (explained in detail in Section 2.3) to estimate the attendance preference of any user u_x to item a_l.

2.2.4 Geographic Closeness Partner Hypothesis. Since most of real-life activities are related to locations, suitable partners for u_t could be the ones who are the closest to the activity location (Ying et al. 2012). For a user u_x, we can define as the user's *geographic footprint* A_x, the set of activity locations most frequently attended by her (i.e., the set of activities visited the largest number of times). Then, the distance from u_x to an activity a_l is modeled as

$$\text{Dist}\left(u^x, a_l\right) = \frac{1}{|A_x|} \sum_{a_x \in A_x} \text{geodist}\left(a_x, a_l\right), \tag{4}$$

where geodist (a_x, a_l) is the geographical distance between a_x's location and a_l's location. For a partner candidate u_x, the together preference is proportional to the geographic closeness from him/her to the activity:

$$pf([u_t, a_l] \to u_x) \propto 1 - \frac{\text{Dist}\left(u^x, a_l\right)}{\max_{u_x} \text{Dist}\left(u^x, a_l\right)}. \tag{5}$$

2.3 Utilizing Historical Together Preferences

In this section, we propose a method to recommend activity partners in a supervised manner (i.e., when historical together preferences are available and can be used for training). Our objective is to predict a user's together preference via his/her past together preference records. We first discuss the possible sources of past together preference data for the target user. Then, we will show how known together preference data can be used to predict together preference for a new item.

2.3.1 Extracting Historical Together Preference Data. Several methods can be used to retrieve together preference data. First, some domains own the together preference data already. For example, consider the case where the activity items are online games. The system that hosts the games can easily record whether two users have played some game together. Together preferences can also be derived from users' behavior at the activity-partner recommendation web service. For example, if we set up an activity-partner recommendation system with an interface similar to the one in Figure 1(3), the clicking behavior of users on the invitation button is an indicator of activity-partner preference. Another source of together-preference data are the check-in records of geo-social networks. Assume that we have access to the check-in data of users together with their social connections. If two users who are friends checked in at the same activity venue very close in time, we can infer that they attended the activity together. For example, two friends who checked in at the same Chinese restaurant at 8:00 p.m. and 8:15 p.m. on the same day, most probably had dinner together.

2.3.2 Employing Historical Together-Preference Data. With the availability of past together-preference data, recommending activity partners seems to be a typical recommendation problem if we regard the combination of target user and activity (e.g., $[u_t, a_l]$) as a special "user." Thus, it seems natural to employ CF for recommending activity partners. However, as we will show next, there are some problems with the direct use of CF to solve our problem. Let us first review how user-based CF (Schafer et al. 2007) works. Suppose that we have to estimate $pf(u_t \to a_l)$ of user u_t on item a_l. The first step of user-based CF is to calculate for each other user u_i the vector similarity between the rating profiles of u_t and u_i (denoted as $\overline{r_t}$ and $\overline{r_i}$), e.g., use Equation (2) to compute the similarity between users. The second step is as follows: if the similarity between user u_t and

u_i (denoted by $S^u_{t,i}$) satisfies some condition (e.g., it is larger than a threshold or in the set of top-k highest similarities), we regard u_i to be in the *neighborhood* of u_t. To predict $p^A_{t,l}$, we aggregate the (known) preference from $p^A_{i,l}$ of all users u_i in the neighborhood of u_t, as follows:

$$af(u_t \to a_l) \propto \frac{\sum_{u_i \in N^t} S^u_{t,i} r_{i,l}}{\sum_{u_i \in N^t} S^u_{t,i}}, \tag{6}$$

where N^t denotes the users in u_t's neighborhood.

Now, assume that we try to apply this conventional user-based CF approach to predict the together preference $pf([u_t, a_l] \to u_x)$. Here, we can regard each $[u_t, a_l]$ as a special user unit and call this "user" unit as *ua-pair*. First, we should try to find the ua-pair neighborhood of $[u_t, a_l]$. However, since activity items recommended to a user should be the ones he/she has not attended yet, we do not have any historical together-preference information for ua-pair $[u_t, a_l]$. This means that all the elements of the profile vector of $[u_t, a_l]$ are unknown, thus we are not able to find neighbor ua-pairs of $[u_t, a_l]$ by computing the vector similarity between the row of $[u_t, a_l]$ and those of other ua-pairs. This problem is not unique to user-based CF. It also occurs when we try to use item-based (Sarwar et al. 2001) or matrix-factorization-based CF (Koren et al. 2009) methods, since the profile row of $[u_t, a_l]$ does not contain any known values.

To solve the problem discussed above, we employ an alternative method for defining the neighbors of $[u_t, a_l]$ and their similarity. We just consider all $[u_t, a_m]$ ($m \neq l$) as candidate neighbor ua-pairs of $[u_t, a_l]$. In other words, we only take the ua-pairs for which the user element is the same as the target user u_t as candidates of neighbor ua-pairs, since we found that the together-preference patterns of different users are very different (this will be demonstrated in the Experiments section). Then, we regard the similarity between $[u_t, a_l]$ and $[u_t, a_m]$ as the similarity between a_l and a_m ($m \neq l$). For example, we can use the similarity between the profile vectors of a_l and a_m (i.e., item similarity) to model the similarity between $[u_t, a_l]$ and $[u_t, a_m]$.[1] After calculating the similarity between $[u_t, a_l]$ and all $[u_t, a_*]$, we select the most similar $[u_t, a_*]$ as the neighbors of $[u_t, a_l]$ (i.e., those with similarity larger than a threshold or those with the highest similarities). Finally, we can predict $pf([u_t, a_l] \to u_x)$ (i.e., $p^x_{t,l}$) by aggregating all together preferences $pf([u_t, a_m] \to u_x)$ (i.e., $p^x_{t,m}$) of $[u_t, a_m]$ ($m \neq l$) on u_x as

$$pf([u_t, a_m] \to u_x) \propto \frac{\sum_{[u_t, a_m] \in \mathcal{N}^{t,l}} S^a_{l,m} p^x_{t,m}}{\sum_{[u_t, a_m] \in \mathcal{N}^{t,l}} S^a_{l,m}}, \tag{7}$$

where $\mathcal{N}^{t,l}$ denotes the neighbor ua-pairs of $[u_t, a_l]$. We denote the above extended CF method by *CFAPR*.

From the above equation, we can see that *CFAPR* actually assumes that people have similar preferences for patterns on similar activities, which is a reasonable assumption. For example, John likes to watch football matches and play football with his sports buddies, but prefers to watch romantic movies and have dinner in a restaurant with his girlfriend. Algorithm 1 summarizes the whole process of *CFAPR*.

3 PARTNER-AWARE ACTIVITY RECOMMENDATION

In Section 2, we have introduced our methods for activity-partner recommendation. Although our activity-partner recommendation model tries to find suitable partners for a user u_t to attend an

[1]Note that the similarity between the target activity and other activities can be calculated by content or geographical-based methods (Zhang and Wang 2015; Yin et al. 2013; Wang et al. 2015) if the target social activity is a one-time event and cold-start item.

ALGORITHM 1: *CFAPR*

Input: (i) $C^{t,l}$: the candidate set of partners recommended to user u_t when recommended activity-related item is a_l; (ii) $S^a(a_l, a_m)$: similarity function between two activity items (i.e., a_l and a_m); (iii) *neighbor_condition*: a threshold or a value k for defining the number of neighbor ua-pairs.
Output: K partners recommended for u_t to attend a_l together

Initial $\mathcal{N}^{t,l} = \emptyset$; $\mathcal{A}^t$ = the activity items previously preferred by u_t;
for all $a_m \in \mathcal{A}^t$ **do** $Sim([u_t, a_l], [u_t, a_m]) = S^a(a_l, a_m)$
for all $a_l \in \mathcal{A}^t$
  if $Sim([u_t, a_l], [u_i, a_m])$ satisfies *neighbor_condition*
  then Add $[u_t, a_m]$ into $\mathcal{N}^{t,l}$
for all $u_x \in C^{t,l}$ **do** Compute $pf([u_t, a_l] \rightarrow u_x)$ using Equation (7)
Return K users in $C^{t,l}$ having the highest K $pf([u_t, a_l] \rightarrow u_*)$ values.

activity together, it is possible that for none of the activity items a_l recommended to u_t we can find partners u_x of high together preference. In this case, we can infer that it is hard for u_x to find a suitable partner to attend any of the activities recommended to her, which increases the likelihood of the activities to be rejected. In general, when people are interested in different activities, they prefer attending the activities for which they can find suitable partners. Therefore, it makes sense to consider the likelihood to find suitable partners when recommending activities to users. In this section, we present *partner-aware activity recommendation* and discuss how conventional recommendation approaches can be adapted and improved to suggest items not only based on the interests of users, but also whether the user can find activity partners.

3.1 Partner-Aware Interest Probability

The objective of partner-aware activity recommendation is to recommend activities that not only arouse the target user's interests, but for which the user can find partners to attend them together. Let us call the probability that a user u is interested in an activity *an interest probability* (denoted by $P(A)$ in this section) from u to a, and the probability that a user u can find partners for attending activity a *partner probability* (denoted by $P(B)$) from u to a. Then, the probability that u is interested in a *and* can find partners for a should be the joint probability $P(AB)$. We call this probability *partner-aware interest probability*. To estimate *interest probability*, we could use previous activity recommendation approaches that predict users' interests to activities based on their historical interests in the activities (i.e., $P(A) \propto af(u \rightarrow a)$). For instance, some methods assume that users like to attend the activities similar with the ones they attended before, or assume that they will attend the activities not far from the place they live. Our question is now how to estimate whether the target user u_t can find a suitable partner to attend some activity together (i.e., *partner probability* $P(B)$). Recall that the together preference $pf([u_t, a] \rightarrow u_c)$ measures how much user u_t likes to attend activity a together with u_c. Assuming that for each item a, we compute the together preferences from user u_t to all partner candidates, the maximum of these together preferences can be used to estimate $pp(u_t \rightarrow a)$. The rationale is that if the top recommended partner to u_t for a has a high together preference, we know that there is at least one suitable partner candidate. Thus, we have *partner probability* $P(B) \propto \max_{u_c} pf([u, a] \rightarrow u_c)$, where u_c is any partner candidate for u to attend a together. Notice that since previous recommenders calculate users' interests to activities without considering whether they can find partners for them, the $P(A)$'s they obtain satisfy $P(A|B) = P(A|\neg B)$. Thus, we have $P(AB) = P(A) \cdot P(B)$. Specifically, for *partner-aware interest probability*, we have $P(AB) \propto (af(u \rightarrow a) \times \max_{u_c} pf([u, a] \rightarrow u_c))$. Finally, we could use *partner-aware interest probability* to rank activities and extract those that u is interested in and

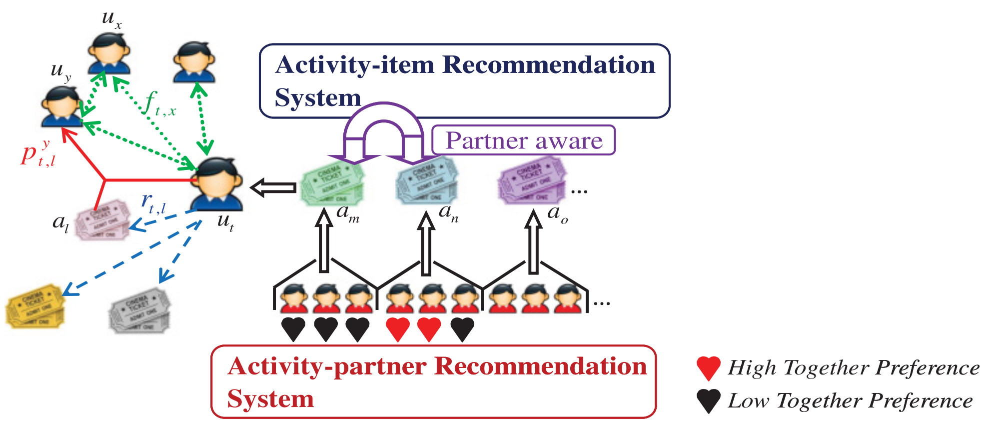


Fig. 3. Illustration of partner-aware recommendation.

could find partners for at the same time (i.e., with high *partner-aware interest probabilities*). Figure 3 shows an example. Suppose the conventional activity recommender sorts activities as a_m, a_n, a_o. We observe that the highest together preference from u_t to partner candidates for a_m is much lower than that for a_n. Our strategy will change the orders of a_m and a_n since the *partner-aware interest probability* from u_t to a_n will be higher than that from u_t to a_m. This ensures that the top recommended item is an activity for which the target user can find partners to attend it together.

3.2 Partner-Aware Activity Recommendation System

The algorithm we propose to implement the above idea is summarized as Algorithm 2. First, for a target user u_t, we use the conventional recommender to predict u_t's *interest probabilities* to all activity-item candidates as $af(u_t \to a_l), a_l \in C^t_{\mathcal{A}}$. Then, activity-item candidates are ranked according to the corresponding *interest probability* (i.e., $af(u_t \to a_l)$). Recall that, after this step, the conventional recommender returns the activity items with the highest k $af(u_t \to a_l)$ values as the recommended activity items, where k is the size of the recommendation list. In our method, before generating the recommendation list, we extract the activity-item candidates with top $\rho * k$ ($\rho > 1$) *interest probabilities* values as a reduced candidate set, denoted as $\mathcal{R}^t_{\mathcal{A}}$. Then, for each activity-item candidate $a_{c'} \in \mathcal{R}^t_{\mathcal{A}}$, we calculate the corresponding *partner probabilities* (denoted as $pp(u_t \to a_{c'})$ in the algorithm) to measure whether u_t can find a partner to attend $a_{c'}$ together. Finally, we obtain the *partner-aware interest probability* as the product of *interest probability* and *partner probability* (i.e., $(af(u_t \to a_{c'}) \times pp(u_t \to a_{c'}))$, which is used to eventually rank the recommendations. Note that we set the minimum value of a partner probability to a very small value $\delta = 1.0e-5$. In other words, if a *partner probability* from the target user to all activity candidates for an item $a_{c'}$ is zero, we adjust it to δ in order for the rank of the item to simply correspond to its attendance preference by the target user. Thus, $a_{c'}$ is ranked w.r.t. $\delta \cdot af(u_t \to a_{c'})$, which would be proportional to $af(u_t \to a_{c'})$, i.e., the relative ranking of $a_{c'}$ compared to other items without suitable partners remains the same. Otherwise, note that our algorithm uses the partner factor to adjust the orders of only the activity candidates with the highest $\rho * k$ attendance preferences rather than all activity candidates. This guarantees that all recommended activities do not have low *interest probability*, considering that *interest probability* is the primary factor in determining whether the user will attend the activity. Moreover, this will decrease computational burden since only the partner probabilities for the top $\rho * k$ activities need to be calculated (this parameter is further discussed in Section 4.4.2). Finally, we generate the recommendation list for u_t as the activities in $\mathcal{R}^t_{\mathcal{A}}$ with the highest k *partner-aware interest probabilities*.

ALGORITHM 2: Partner-Aware Activity Recommendation

Require:

u_t: target user set
$C^t_{\mathcal{A}}$: the candidate set of activity-related items recommended to u_t
$C^{t,l}_{\mathcal{P}}$: the candidate set of partners recommended for u_t to attend activity a_l together
$af(u_t \rightarrow a_l)$: attendance preference estimator (e.g., Equation (6))
$pf([u_t, a_l] \rightarrow u_p)$: together preference estimator (e.g., Equation (7))
k and ρ: parameters

Ensure:

Recommendation list with k activities for the target user u_t

for all $a_c \in C^t_{\mathcal{A}}$ **do**
  calculate attendance preference $af(u_t \rightarrow a_c)$
end for
$\mathcal{R}^t_{\mathcal{A}}$ = elements in $C^t_{\mathcal{A}}$ with highest $\rho * k$ attendance preferences.
for all $a_{c'} \in \mathcal{R}^t_{\mathcal{A}}$ **do**
  $pp(u_t \rightarrow a_{c'}) = \delta$
  for all $u_p \in C^{t,c'}_{\mathcal{P}}$ **do**
    calculate together preference $pf([u_t, a_{c'}] \rightarrow u_p)$
    if $pf([u_t, a_{c'}] \rightarrow u_p) > pp(u_t \rightarrow a_{c'})$ **then**
      $pp(u_t \rightarrow a_{c'}) = pf([u_t, a_{c'}] \rightarrow u_p)$
    end if
  end for
  calculate partner-aware interest probability
  $paf(u_t \rightarrow a_{c'}) = af(u_t \rightarrow a_c) \times pp(u_t \rightarrow a_{c'})$
end for
Recommended list = items in $\mathcal{R}^t_{\mathcal{A}}$ having the highest k $paf(u_t \rightarrow a_{c'})$ values.

4 EXPERIMENTS

In this section, we evaluate the effectiveness of our work in recommending activity partners and improving activity recommendation. Section 4.1 demonstrates the meaningfulness of activity-partner recommendation via feedback collected from real Web users. Section 4.2 introduces the datasets we used and the setup of our experiments. Section 4.3 evaluates the performance of several strategies on recommending activity partners. Section 4.4 compares the performance of traditional recommendation methodologies and our partner-aware strategy.

4.1 Users' Favor of Activity-Partner Recommendation

To confirm the practical value of our work, we conducted an electronic survey (Root and Draper 1983) that involved real-world Web users. The objective is to find out whether users to whom activity items are recommended are also interested in activity-partner recommendation for these items. The designed questionnaire asks people whether they prefer to receive activity-partner recommendations together with the corresponding activity items and was released to public Chinese Web users since November 21, 2014. Until the submission of this work, 197 Web users (from various provinces of China) returned their answers to us. Although we did not get much feedback (there were very few Web users willing to fill in the online questionnaire without a reward), we

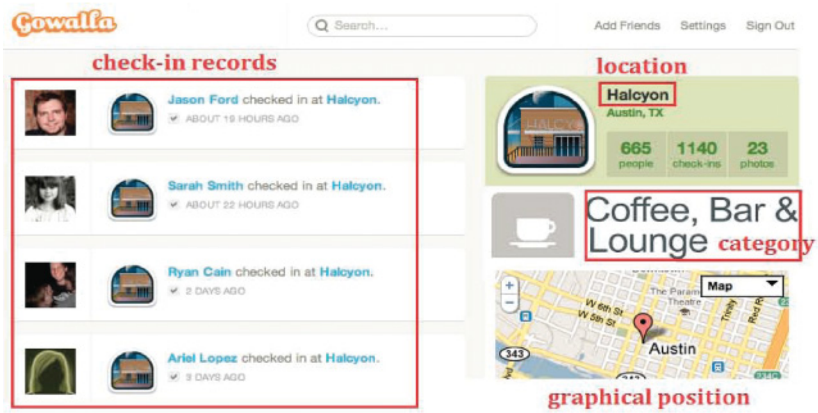


Fig. 4. Typical LBSN datasets (screenshot from the Gowalla website).

believe that the sample is big enough to reflect the opinion of typical Web users. Finally, **about 93.4% of participating users expressed their preference to activity-partner recommendation, compared to recommending activity items alone.** This indicates that our study has good potential in improving the quality of current recommender systems.

4.2 Datasets and Experimental Setup

In our effectiveness evaluation, we used data from location-based social networks to simulate a real-world scenario (i.e., social-activity-based platform) for our work. We first give a brief introduction of the datasets we used. Then, we discuss how we use them to simulate a social-activity-based platform and how to obtain ground truth for the activity partners. Finally, we present our experimental setup.

4.2.1 Datasets. Figure 4 shows screenshots of Gowalla, which is a typical location-based social network (LBSN) that collects check-in records. A check-in record (u_t, l_i, t) states that user u_t checked in at location l_i at time t. For example, the screenshot shows several users' check-in records at Halcyon, which is a place in the Coffee, Bar & Lounge category. In our experiments, we used data from two public LBSNs: Foursquare (Gao et al. 2012) and Gowalla (Cho et al. 2011). The Foursquare dataset includes 1,385,223 check-in records from 11,326 users to 182,968 locations from January 1, 2011 to July 31, 2011. The Gowalla dataset includes 6,442,892 check-in records from 107,092 users to 1,280,969 locations, from February 4, 2009 to October 23 2010. Both datasets have check-in timestamps and social links between users. Since the experiments involve a training period and a test period, we split each of the two LBSN datasets into a training set and test set in the same manner as in Gao et al. (2012) and Cho et al. (2011). For Foursquare data, we use the check-in data ranging from January 1 to June 30 as the training set to learn our model parameters, and construct the testing set from the check-in data in July. For Gowalla data, we put 80% of the check-ins in the training set and the remaining 20% in the test set.

4.2.2 Social-Activity-Related Locations. We regard locations in LSBN datasets as social-activity items (i.e., $l_i = a_i$). This is reasonable, since many activity items (e.g., tickets, dinner vouchers) refer to particular locations at particular time periods or moments. In order for our experiments to focus on locations that can be mapped to social activities, we use the categories of locations to characterize the social-activity locations. However, the public versions of Foursquare (Gao et al. 2012) and

(a) On Foursqure. (b) On Gowalla.

Fig. 5. Words appearing frequently in location categories.

Table 1. Social-Activity Keywords Used in Our Experiments

Social-activity keywords
gym, restaurant, food, cafe, bar, BBQ, club, spot, coffee, entertainment, fitness, art, museum, opera, steakhouse, plaza, bookstore, motel, hotel, beer, football, basketball, music, mall, bike, theater, movie, nightlife, park, yoga, burger, sandwich, sport, sushi

Table 2. Examples of Location Categories

Examples of location categories related to social activities	Examples of location categories not related to social activities
xinjiang restaurant, water park, gaming cafe, wine bar, college gym, steakhouse, art museum, sports club, movie theater, basketball court, gay bar, turkish restaurant, juice bar, theme park ride attraction, beer garden	airport tram, light rail, tunnel, train station, paper office supplies store, financial or legal service, bus stop, costume shop, monastery, track, middle school, doctor's office, factory, real estate office

Gowalla (Cho et al. 2011) datasets only contain check-in records and friend links between users, but no category information about the places. Thus, we crawled the category information of all locations (in both Foursquare and Gowalla data) using Foursquare API (developer.foursquare.com). The API outputs the category of a location if we give its longitude and latitude as input. Since both Foursquare and Gowalla datasets contain the longitudes and latitudes of locations, we successfully obtained category information of all locations in them. Figure 5 shows the most frequent words appearing in the location categories of the two datasets. As we can see, the most frequent word in the location categories of Foursquare and Gowalla datasets is *restaurant*. The restaurant locations can be easily mapped to social activity "having dinner at a restaurant" and activity-related items like "discount coupon for dinner." However, note that there are some categories (e.g., *store* and *station*) which are hard to map to social activities. Thus, to make the experiments focus on the locations related to social activities, we filter out the locations in the raw datasets that are hard to relate to any social activity. Specifically, we only keep the records related to the locations of which the categories contain social-activity-related keywords (e.g., *restaurant*, *bar*, *gym*). The whole list of social-activity-related keywords we used is shown in Table 1. Several examples of location categories related to (and not related to) social activities are given in Table 2 to verify the effectiveness of the keywords in finding social-activity-related locations. Using location categories related to social activities, about one-half of locations are mapped to social activities. Specifically, 90,803 from 182,968 Foursquare locations and 625,632 Gowalla locations are mapped to social activities. Then, activity-partner recommendation and activity recommendation are performed based on the check-in records related to social activities. There are 598,345 and 2,976,367 check-in records related to social-activity locations in Foursquare and Gowalla datasets, respectively.

4.2.3 Ground Truth of Activity Partners. Besides the fact that the locations can be mapped to social activities, another reason for which we used LBSN datasets in our experiments is that we can simulate the ground truth of activity partners from them. In an LBSN, we know the timestamps of check-ins and the friendship links between users; based on these, we can infer the ground truth about activity partners. Here we employed the method used in Purushotham et al. (2014) for extracting group members to obtain the ground truth about activity partners. Specifically, as discussed in Section 2.3, if two users u_x and u_y are friends and check in at a same location (activity) a_i at close timestamps (i.e., the time difference between their check-in timestamps is less than a threshold $\mathcal{T}$), we regard that the two users attended the corresponding activity item (location) together (denoted as $([u_x, a_i], u_y)$ and thus they are activity partners of each other with respect to the activity item a_i. In other words, we have $([u_x, a_i], u_y)$ if there exist (u_x, a_i, t_1) and (u_y, a_i, t_2) in the LBSN data, $|t_1 - t_2|$ is less than $\mathcal{T}$, and $f_{x,y} = 1$. In our experiments, we set $\mathcal{T}$ as 3 hours since most of real-world activities mapped to activity-related locations last about 3 hours. For example, the typical duration of a movie is between 2 and 3 hours, therefore the activity "watching a movie" typically lasts 2 to 3 hours; i.e., if two friends watch a movie together, the difference between their check-in time stamps could be from zero to 3 hours. We have analyzed the influence of this threshold on constructing ground truth of activity partners. There are no significant differences among the sets of activity partners corresponding to setting the threshold as 1, 2, 3 hours since most of the check-in time differences between activity partners are less than 1 hour. Thus, the influence of this threshold on the experimental results is limited.

4.2.4 Recommendation Problems and Evaluation Metrics. Given each pair of user u_t and activity a_l, the activity-partner recommendation problem is to suggest to u_t a list of users, with the expectation that the recommended users $u_1, u_2, \ldots, u_K$ contain real activity partners for $[u_t, a_l]$. Note that according to the rule of extracting ground truth of activity partners, a real partner must be linked with the target user in the social network. Thus, the partner candidates in our activity-partner recommendation experiments are set as all users who are socially linked with target users. In order to evaluate the performance of recommending activity partners, we use the classic precision and recall metrics (Gunawardana and Shani 2009; Bao et al. 2015):

$$Precision = \frac{\sum_{[u_t, a_l] \in \mathcal{V}} |Pa^{rec}([u_t, a_l]) \bigcap Pa^{real}([u_t, a_l])|}{\sum_{([u_t, a_l]) \in \mathcal{V}} |Pa^{rec}(u_t, a_l])|}, \tag{8}$$

$$Recall = \frac{\sum_{[u_t, a_l] \in \mathcal{V}} |Pa^{rec}([u_t, a_l]) \bigcap Pa^{real}([u_t, a_l])|}{\sum_{([u_t, a_l]) \in \mathcal{V}} |Pa^{real}([u_t, a_l]])|}. \tag{9}$$

The test user-activity pair set $\mathcal{V}$ contains all $[u_t, a_l]$ pairs for which there is at least one activity partner in our ground-truth data. $Pa^{real}([u_t, a_l])$ includes the actual activity partners of $[u_t, a_l]$ and $Pa^{rec}([u_t, a_l])$ is the set of recommended partners generated by an evaluated activity-partner recommender. Similarly, when we evaluate our partner-aware activity recommender against baselines, we also employ precision and recall as evaluation metrics:

$$Precision = \frac{\sum_{u_t \in \mathcal{U}} |At^{rec}(u_t) \bigcap At^{real}(u_t)|}{\sum_{(u_t) \in \mathcal{U}} |At^{rec}(u_t)|}, \tag{10}$$

$$Recall = \frac{\sum_{u_t \in \mathcal{U}} |At^{rec}(u_t) \bigcap At^{real}(u_t)|}{\sum_{(u_t) \in \mathcal{U}} |At^{real}(u_t)|}. \tag{11}$$

Here, $\mathcal{U}$ is the set of test users, $At^{real}(u_t)$ consists of the activities attended by u_t, and $At^{rec}(u_t)$ is the recommended activity list generated by some activity recommender.

4.2.5 Conventional Activity Recommenders. In our experiments, we employed several general-purpose recommendation approaches as modules of our method and for comparison purposes. First, we use them to calculate $af(u \rightarrow a)$ for ALAPR (see Equation (15)). Second, they are employed for attendance preference estimation in our partner-aware activity recommendation framework (see Algorithm 2). When each of the activity recommenders is used in the partner-aware framework, we obtain its corresponding partner-aware version. Thus, we will compare the performances between each pair of conventional activity recommender and its partner-aware version to demonstrate the effectiveness of our partner-aware activity recommendation framework. In our work, we select six conventional activity recommenders commonly used in previous work on two experimental datasets.

B$_1$: User-based CF (UCF). The most popular approach in recommender systems is User-based Collaborative Filtering (used in Chow et al. (2010), Ye et al. (2010), Ye et al. (2011), and Horozov et al. (2006)), where recommendations are created based on the past behavior of a user. UCF assumes that similar users have similar preferences over items. As we introduced in Section 2.3, the attendance preference $af(u_t \rightarrow a_l)$ is computed by the average of other user ratings on a_l, weighted by the similarity of these users to u_t:

$$af(u_t \rightarrow a_l) \propto \frac{\sum_{u_i \in N^t} S^u_{t,i} r_{i,l}}{\sum_{u_i \in N^t} S^u_{t,i}}, \tag{12}$$

where $S^u_{t,i}$ is the similarity between users u_t and u_i, most commonly defined as the cosine similarity between the preference profiles of the users, i.e., $S^u_{t,i} = \cos(\overline{r_t}, \overline{r_c})$ and N^t denotes the users in u_t's neighborhood.

B$_2$: Item-based CF (ICF). Item-based Collaborative Filtering (used in Wang et al. (2013)) comes from the assumption that people like similar items. To estimate the preference of user u_t on activity a_l, instead of comparing the user vectors as in UCF, we compare the profile (i.e., vector) of a_l to the profiles of other items. The attendance preference $af(u_t \rightarrow a_l)$ in ICF is

$$af(u_t \rightarrow a_l) \propto \frac{\sum_{a_m \in A^u} S^a_{l,m} r_{t,m}}{\sum_{a_m \in A^u} S^a_{l,m}}, \tag{13}$$

where A^u consists of the activities rated by u_t before and $S^a_{l,m}$ measures the similarity between a_l and a_m. Similar to UCF, $S^a_{l,m}$ could be the cosine similarity $\cos(\boldsymbol{r}^a_l, \boldsymbol{r}^a_m)$, where $\boldsymbol{r}^a_l$ and $\boldsymbol{r}^a_m$ are the vectors formed by the preferences of all users to a_l and a_m, respectively.

B$_3$: Friend-based CF (FCF). Friend-based CF (used in Ye et al. (2010), Ye et al. (2011), Wang et al. (2013), and Scellato et al. (2011)) considers only friends when applying collaborative filtering for a target user. FCF assumes that people listen to their friends and follow their friends' recommendations only. As such, FCF only needs to compute the similarities between the given user and her friends only, instead of all users. Since non-friends are not considered, noise by users who are not expected to influence the target user is reduced, which is expected to improve the precision of the recommendations. FCF calculates $af(u_t \rightarrow a_l)$ as

$$af(u_t \rightarrow a_l) \propto \frac{\sum_{u_i \in F^t} S^u_{t,i} r_{i,l}}{\sum_{u_i \in F^t} S^u_{t,i}}, \tag{14}$$

where F^t is the set of u_t's friends.

B$_4$: Social-network-based CF (SCF). Similar to FCF, social-based CF (used in Ye et al. (2010), Ye et al. (2011), and Wang et al. (2013)) utilizes friendship information for recommending items. Different from FCF which considers only u_t's direct friends $\mathcal{F}^t$ as the set of similar users to u_t, SCF calculates similarity between users by the set of their common friends:

$$af(u_t \rightarrow a_l) \propto \frac{\sum_{u_i \in N^t} S^u_{t,i} r_{i,l}}{\sum_{u_i \in N^t} S^u_{t,i}}, \tag{15}$$

where user similarity $S^u_{t,i} = \text{Jaccard}\left(\mathcal{F}^u, \mathcal{F}^i\right)$, and Jaccard $(\cdot,\cdot)$ is the Jaccard index.

B$_5$: Geo-distance-based CF (GCF). The rationale of Geo-distance-based CF (used in Horozov et al. (2006), Ying et al. (2012), and Ye et al. (2011)) is that nearby friends are more influential than faraway ones. Let us denote the locations that user u and v most frequently attended as L^u and L^v, respectively. Then, the distance between u and v (denoted as Dist (u, v)) is estimated as the average value of the distances between location pairs (l_u, l_v) for all $l_u \in L_u$, $l_v \in L_v$. Finally, the similarity between user u_t and u_i is calculated as

$$S^u_{t,i} = 1 - \frac{\text{Dist}\,(u_t, u_i)}{\max_{u_j} \text{Dist}\left(u_t, u_j\right)}. \tag{16}$$

B$_6$: Category-based CF (CCF). Category-based CF (used in Bao et al. (2012) and Xiao et al. (2010)) considers users as keywords and location categories as documents; between user u and category C there can be a *relevance score*, rel(u, C) (e.g., TF-IDF). To measure the similarity between two users u and i, the sum of minimum relevance scores over all categories, i.e., $\text{sim}(u, v) = \sum_C \min\{\text{rel}(u, C), \text{rel}(v, C)\}$ is used. This value is then penalized by the difference between users' *randomness* in preferences, thus the weight $S^u_{t,i}$ is

$$S^u_{t,i} = \frac{\text{sim}(u, I)}{1 + |\text{ent}(u) - \text{ent}(i)|}, \tag{17}$$

where ent$(\cdot)$ is the *entropy* of a user's preference over categories.

4.3 Effectiveness of Activity-Partner Recommenders

In this section, we will compare the performance of the activity-partner recommendation approaches introduced in Section 2.

4.3.1 Competitors. Recall that for the activity-partner recommendation problem, which is studied in this article for the first time, we have introduced several hypotheses about the preferences of users on potential activity partner. SCAPR, SIAPR, ALAPR, and GCAPR are proposed in Section 2.2 for performing activity-partner recommendation, by considering the users' social context, attendance behaviors, and geo-information. Moreover, we adapted the traditional collaborative filtering model into a CFAPR model in Section 2.3, in order to recommend activity partners in a CF manner. Here, for the sake of experimental evaluation, we include a competitor to CFAPR, which also applies training using historical together preferences: *Popular-Partner-based APR (PPAPR)* models the popularity of an activity partner candidate by the times s/he is preferred as an activity partner by the target user. In other words, PPAPR is based on a *partner consistency* hypothesis while for

Table 3. A Summary of Evaluated Methods

Method	Principle
CFAPR	Assuming that users prefer to attend activities with those who are activity partners on similar activities, *CFAPR* ranks the activity-partner candidates by $\hat{p}^c_{t,l}$ calculated by Equation (7), where c corresponds to a partner candidate u_c.
PPAPR	Assuming that users prefer to attend activities with those who they usually attend other activities with, *PPAPR* ranks the activity-partner candidates by $Pop(u_t, u_c)$ calculated by Equation (18), where c corresponds to a partner candidate u_c.
SIAPR	Assuming that users prefer to attend activities with those who have similar interests with them, *SIAPR* ranks the activity-partner candidates by $S^u_{t,c}$, calculated by Equation (2), where c corresponds to a partner candidate u_c.
SCAPR	Assuming that users prefer to attend activities with those who are socially close to them, *SCAPR* ranks the activity-partner candidates by $SC(u_t, u_c)$, calculated by Equation (3), where c corresponds to a partner candidate u_c.
ALAPR	Assuming that users prefer to attend activities with those who also prefer the activity item, *ALAPR* ranks the activity-partner candidates by estimated interest from the partner candidate to the activity $af(u_c \rightarrow a)$ (see Equation (15)), where c corresponds to a partner candidate u_c.
GCAPR	Assuming that users prefer to attend activities with those who usually attend the activities near the target activity, *GCAPR* ranks the activity-partner candidates by the distance between the location of the target activity and the partner candidate's most frequently visited location, calculated by Equation (5), where c corresponds to a partner candidate u_c.

the CFAPR model of Section 2.3 the choice of a partner mostly depends on the activities. In PPARP, the popularity of a partner candidate for a target user is defined as

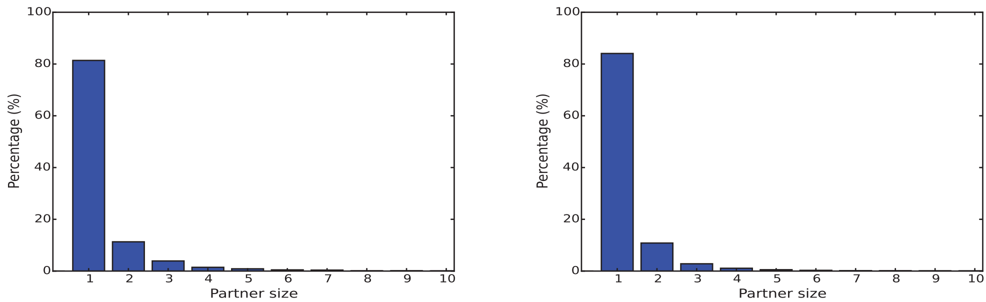
$$pf([u_t, a_l] \rightarrow u_c) \propto Pop(u_t, u_c) = |\mathcal{V}^t_c|, \tag{18}$$

where $\mathcal{V}^t_c$ is the set of valid user-activity pairs of user u_t whose activity partners include u_c. The evaluated methods are summarized in Table 3.

Note that prior knowledge of together preferences is a requirement for methods PPAPR and CFAPR. Thus, for both these methods, we split the test users into two sets. One contains the users having past together behaviors in the training set, and the other consists of the remaining users. We call the first set *warm users* and the latter *cold users*.

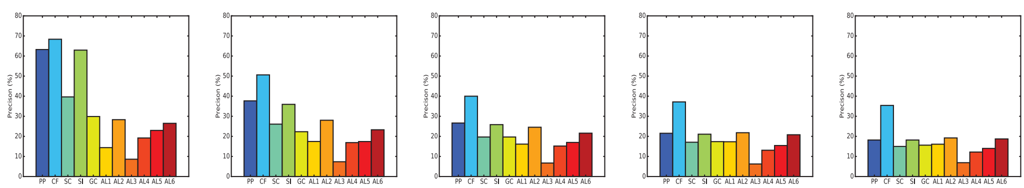
4.3.2 Results and Analysis. Considering that in real-world applications, the size of recommendation window shown to users is limited, it is impossible to recommend many partner candidates for each of the promoted activities if we like to recommend several activities to the target user simultaneously. Moreover, the number of partners that a user may select for an activity is typically small (see Figure 6 for details). Therefore, the list of recommended partners to a user should not be large. Considering all the above, in our experiments, we set the number of recommender activity items K to range in $\{1, 2, 3, 4, 5\}$. Figure 7 shows the results of all APR methods for recommending activity partners to warm users.

Note that since we have six conventional activity recommenders for calculating $af(u \rightarrow a)$, we have five versions of ALAPR: AL1, AL2, AL3, AL4, AL5, and AL6 indicates ALAPR with UCF, ICF, FCF, SCF, GCF, and CCF, respectively. When comparing the performance of different methods, we can observe that

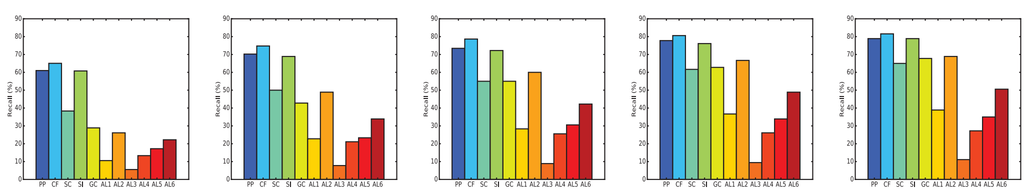


(a) On Foursqure. (b) On Gowalla.

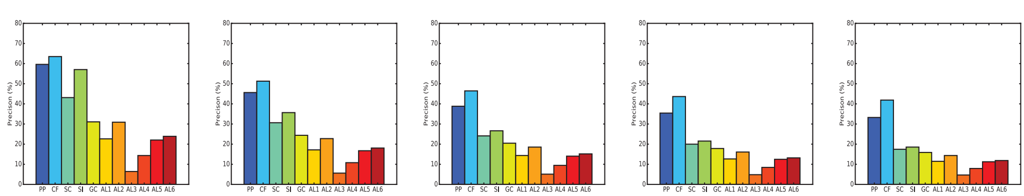
Fig. 6. The distribution of partner size in Foursquare (left) and Gowalla (right) datasets.



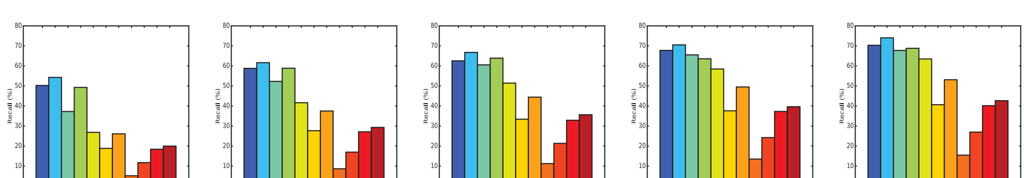
(a) Precison: K=1,Foursquare (b) Precison: K=2,Foursquare (c) Precison: K=3,Foursquare (d) Precison: K=4,Foursquare (e) Precison: K=5,Foursquare



(f) Recall: K=1,Foursquare (g) Recall: K=2,Foursquare (h) Recall: K=3,Foursquare (i) Recall: K=4,Foursquare (j) Recall: K=5,Foursquare



(k) Precison: K=1, Gowalla (l) Precison: K=2, Gowalla (m) Precison: K=3, Gowalla (n) Precison: K=4, Gowalla (o) Precison: K=5, Gowalla



(p) Recall: K=1, Gowalla (q) Recall: K=2, Gowalla (r) Recall: K=3, Gowalla (s) Recall: K=4, Gowalla (t) Recall: K=5, Gowalla

Fig. 7. Performance comparison of methods *CFAPR*, *PPFAPR*, *SCAPR*, *SIAPR*, *GCAPR*, and *ALAPR* for warm users.

— *CFAPR outperforms all other methods.* This indicates the suitability of *CFAPR* for activity-partner recommendation with training together-preference knowledge.
— *CFPAR outperforms PPARP.* Both *CFAPR* and *PPARP* make use of past together preferences. The difference is that *CFAPR* assumes that the together preferences of a user on similar activity items are similar. The fact that *CFAPR* outperforms *PPARP* confirms the validity of this assumption.
— *CPRAPR and PPARP outperform SIAPR, SCAPR, ALAPR.* In general, the methods which use past together preferences (i.e., *CFAPR* and *PPARP*) of the target user perform better than methods which ignore this parameter (i.e., *SIAPR*, *SCAPR*, *ALAPR*). This fact shows that past together preferences play an important role in predicting activity partners.
— *SIAPR outperforms SCAPR, ALAPR. SIAPR*, *ALAPR*, and *SCAPR* are three methods which use information commonly seen in e-commerce or LBSN websites. Exploring their performance can pave the way toward constructing an initial activity-partner recommender for the case where there is no past partner knowledge about the target user. As the results show, *SIAPR* performs best among these three simple methods. Therefore, when there is no training together-preference knowledge, *SIAPR* is a good choice to start up an activity-partner recommendation system.

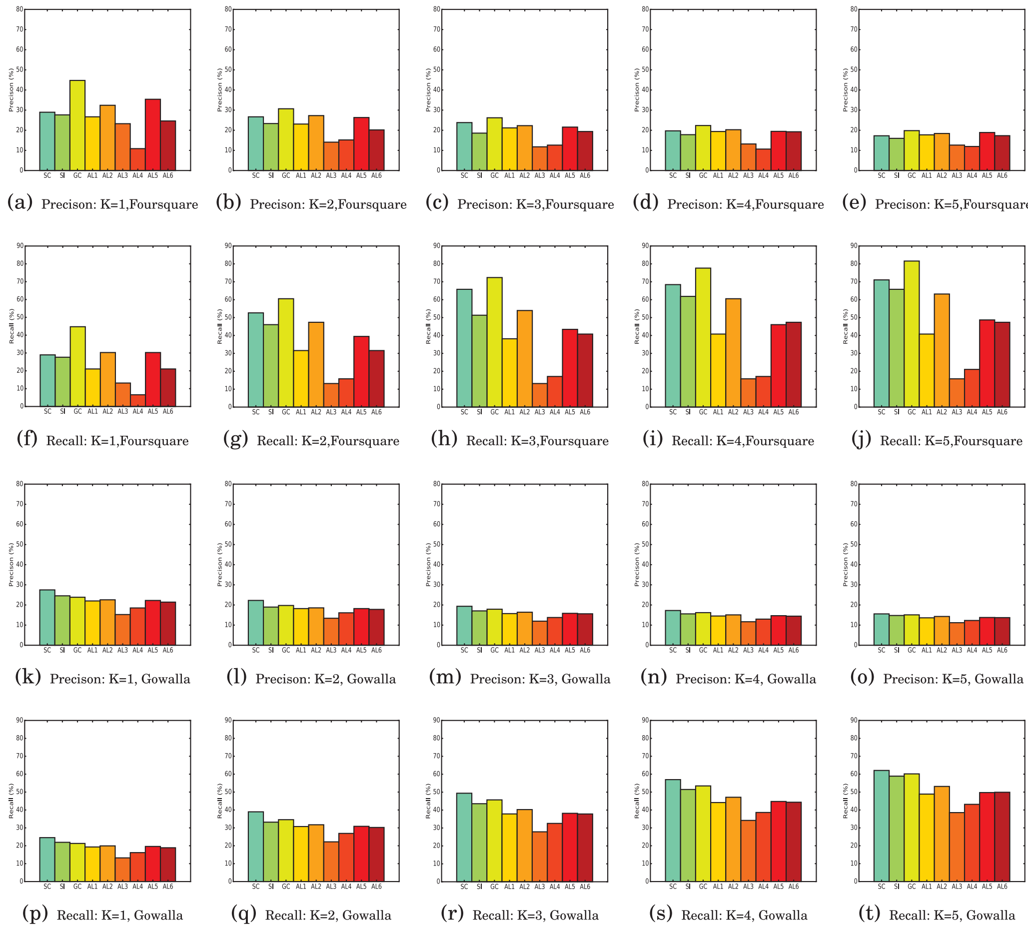
Note that results in Figure 7 are on warm users; for those users with no prior knowledge of together preferences (i.e., cold users), we show the recommendation results in Figure 8. From the results, we see that SIAPR performs poorly while SCAPR and GCAPR perform well. This may be because cold users have no past together preference. We find that these users' attendance history is not as rich as the warm users' attendance history (since a user who has rich attendance history tends to have together preference records). Therefore, SCAPR and GCAPR which use social and geographic information could alleviate the problem of data sparsity and perform relatively well (Chen et al. 2012). **Summing up, for activity-partner recommendation, CFAPR is a good choice for the users having prior knowledge of together preference, while GCAPR and SCAPR are appropriate for users without past together preferences.**

4.4 Effectiveness of Partner-Aware Activity Recommendation

In the next set of experiments, we verify the effectiveness of the partner-aware framework for recommending activities to users. The partner-aware recommendation framework improves conventional recommendation methods, by using the attendance preferences and taking partner probability into consideration. By employing each of the baseline approaches introduced in Section 4.2.5, we obtain their partner-aware versions: Partner-Aware User-based CF (PAUCF), Partner-Aware Item-based CF (PAICF), Partner-Aware Friend-based CF (PAFCF), Partner-Aware Social-network-based CF (PASCF), Partner-Aware Geo-distance-based CF, and Partner-Aware Category-based CF (PACCF).

4.4.1 Assumption Verification. Before we present the results of recommending activities, we use the experimental datasets to verify the assumption of our partner-aware activity recommendation: *among the activities for which a user has similar attendance preferences, the user tends to select the ones for which she can find partners (i.e., high partner probability).*

For this purpose, we first simulate the following scenario: a user (e.g., u) has approximately the same attendance preference to a set of activities. We such an activity set as u's *Close Attendance Preference (CAP)* activity set. Specifically, if an activity set $A = \{a_1, \ldots, a_n\}$ is a *CAP* activity set of u, we should have $af(u \rightarrow a_1) \approx af(u \rightarrow a_2) \approx \cdots \approx af(u \rightarrow a_n)$. In order to identify *CAP* activity sets, we employ the experimental datasets introduced in Section 4.2.1, and conventional activity recommenders (for estimating attendance preferences) introduced in



(a) Precison: K=1,Foursquare (b) Precison: K=2,Foursquare (c) Precison: K=3,Foursquare (d) Precison: K=4,Foursquare (e) Precison: K=5,Foursquare

(f) Recall: K=1,Foursquare (g) Recall: K=2,Foursquare (h) Recall: K=3,Foursquare (i) Recall: K=4,Foursquare (j) Recall: K=5,Foursquare

(k) Precison: K=1, Gowalla (l) Precison: K=2, Gowalla (m) Precison: K=3, Gowalla (n) Precison: K=4, Gowalla (o) Precison: K=5, Gowalla

(p) Recall: K=1, Gowalla (q) Recall: K=2, Gowalla (r) Recall: K=3, Gowalla (s) Recall: K=4, Gowalla (t) Recall: K=5, Gowalla

Fig. 8. Performance comparison of methods *SCAPR*, *SIAPR*, *GCAPR*, and *ALAPR* for cold users.

Section 4.2.5. For each user u in the test set, we use the trained activity recommender to calculate attendance preferences from u to all activities. After that, we rounded the attendance preferences by only keeping the two most significant digits. Then, the attendance preferences are regarded as being approximately the same, if their smoothed values are the same. For example, suppose the attendance preferences from user u to five activities a_1, a_2, a_3, a_4, a_5 are 0.024983759843275, 0.02489789798, 0.027765765, 0.00198798798, 0.00198375843. We will have $af(u \rightarrow a_1) \approx af(u \rightarrow a_2)$ and $af(u \rightarrow a_4) \approx af(u \rightarrow a_5)$. Thus, we obtain two *CAP* activity sets, each consisting of activities with similar attendance preference. In our example, we will have *CAP* activity sets $A_1 = \{a_1, a_2\}$ and $A_2 = \{a_4, a_5\}$. Assume that, after using all test users' data, we have N CAP activities sets, denoted as $A_1, A_2, \ldots, A_N$.

The next step is that, in each A in $\{A_1, A_2, \ldots, A_N\}$, we extract activities with relatively higher partner probabilities to generate a subset A^h and the ones with lower partner probabilities to form the subset A^l. Specifically, we calculate $pp(u \rightarrow a)$ for all $a \in A$. Then we take the activities having top m% (e.g., $m = 30$) $pp(u \rightarrow a)$ values to generate A^h and the ones with bottom m% (e.g., $m = 30$) to form the set A^l.

Now, we have N *CAP* activity sets $A_1, A_2, \ldots, A_N$ for user u. In each of these sets $A = \{a_1, \ldots, a_n\}$, we have $af(u \rightarrow a_1) \approx af(u \rightarrow a_2) \approx \cdots \approx af(u \rightarrow a_n)$. Meanwhile, each

A is divided into two disjoint subsets A^h and A^l, as described in the previous paragraph. If our assumption is true, user u should tend to attend the activities in A^h, compared with activities in A^l. To verify this assumption, we compare the percentage of activities in A^h and in A^l that u indeed attended. We expect the first number (denoted as $P(u \rightarrow a|a \in A^h)$) to be significantly larger than the second (denoted as $P(u \rightarrow a|a \in A^l)$). Specifically, when we have a *CAP* activity set A, we compare

$$P(u \rightarrow a|a \in A^h) = \frac{|u \rightarrow a_i, a_i \in A^h|_i}{|a_i \in A^h|_i} = \frac{|A_u^h|}{|A^h|}, \tag{19}$$

$$P(u \rightarrow a|a \in A^l) = \frac{|u \rightarrow a_j, a_j \in A^l|_j}{|a_j \in A^l|_j} = \frac{|A_u^l|}{|A^l|}, \tag{20}$$

where $|A_u^i|(|A_u^j|)$ is the number of the activities in $A^h(A^l)$ which were attended by u.

Note that we have N *CAP* activity sets. We calculate the average $P(u \rightarrow a|a \in A^h)$ (denoted as $\overline{P^h}$), average $P(u \rightarrow a|a \in A^l)$ (denoted as $\overline{P^l}$), and the percentage of the cases satisfy $P(u \rightarrow a|a \in A^h) > P(u \rightarrow a|a \in A^l)$ (denoted as $P^{h>l}$) as

$$\begin{gathered}\overline{P^h} = \frac{1}{N}\sum_{i=1}^{N} P(u \rightarrow a|a \in A_i^h), \qquad \overline{P^l} = \frac{1}{N}\sum_{i=1}^{N} P(u \rightarrow a|a \in A_i^l), \\ P^{h>l} = \frac{1}{N}\sum_{i=1}^{N}(P(u \rightarrow a|a \in A_i^h) > P(u \rightarrow a|a \in A_i^l)).\end{gathered} \tag{21}$$

Moreover, to explicitly show the results are under the scenario "the activities in A^h and A^l have close attendance preferences but different partner probabilities," we also use measurements to evaluate the differences among attendance preferences in A^l and A^h (denoted as D_{af}^{hl}), as well as the differences among partner probabilities in A^l and A^h (denoted as D_{pp}^{hl}):

$$D_{af}^{hl} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\overline{af}(A_i^h) - \overline{af}(A_i^l)|}{|\overline{af}(A_i^h)| + |\overline{af}(A_i^l)|}, \qquad D_{pp}^{hl} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\overline{pp}(A_i^h) - \overline{pp}(A_i^l)|}{|\overline{pp}(A_i^h)| + |\overline{pp}(A_i^l)|}, \tag{22}$$

where

$$\begin{gathered}\overline{af}(A_i^h) = \frac{1}{|A_i^h|}\sum_{a \in A_i^h} af(u \rightarrow a), \qquad \overline{af}(A_i^l) = \frac{1}{|A_i^l|}\sum_{a \in A_i^l} af(u \rightarrow a), \\ \overline{pp}(A_i^h) = \frac{1}{|A_i^h|}\sum_{a \in A_i^h} pp(u \rightarrow a), \qquad \overline{pp}(A_i^l) = \frac{1}{|A_i^l|}\sum_{a \in A_i^l} pp(u \rightarrow a).\end{gathered} \tag{23}$$

Recall that we have six attendance-preference estimators. We will show the results of using each of them to predict $af(u \rightarrow a)$. According to the experimental results in Section 4.3, CFAPR is the best method to evaluate $pf([u,a] \rightarrow u_c)$. Therefore, we employ it to calculate together preferences and obtain partner probabilities. Moreover, recall that when we generate A^h and A^l, we use a parameter m. Here m varies from 10 to 50.

Table 4 represents the final values of D_{af}^{hl}, D_{pp}^{hl}, $\overline{P^h}$, $\overline{P^l}$ and $P^{h>l}$ based on the above experimental setup. From the results, we can see that

— D_{af}^{hl} is very small while D_{pp}^{hl} is much larger. This confirms that the experiment is consistent with the scenario that "the activities in A^h and A^l have approximately the same attendance preferences but different partner probabilities." Since A^h consists of the activities with top $m\%$ partner probabilities and A^l is formed by the activities with bottom $m\%$ partner

Table 4. Results for Assumption Verification

m%	UCF predicts $af(u \rightarrow a)$										ICF predicts $af(u \rightarrow a)$									
	On Foursquare					On Gowalla					On Foursquare					On Gowalla				
	D^{hl}_{af}	D^{hl}_{pp}	$\overline{P^h}$	$\overline{P^l}$	$P^{h>l}$	D^{hl}_{af}	D^{hl}_{pp}	$\overline{P^h}$	$\overline{P^l}$	$P^{h>l}$	D^{hl}_{af}	D^{hl}_{pp}	$\overline{P^h}$	$\overline{P^l}$	$P^{h>l}$	D^{hl}_{af}	D^{hl}_{pp}	$\overline{P^h}$	$\overline{P^l}$	$P^{h>l}$
10%	0.004	0.84	0.23	0.09	0.77	0.008	0.79	0.22	0.06	0.74	0.008	0.83	0.19	0.06	0.75	0.010	0.83	0.15	0.05	0.71
20%	0.004	0.80	0.19	0.09	0.72	0.009	0.77	0.21	0.07	0.71	0.008	0.84	0.16	0.06	0.74	0.009	0.12	0.83	0.05	0.70
30%	0.004	0.77	0.17	0.09	0.69	0.009	0.75	0.20	0.07	0.69	0.007	0.78	0.14	0.05	0.72	0.009	0.82	0.11	0.04	0.69
40%	0.004	0.74	0.15	0.09	0.66	0.008	0.72	0.19	0.07	0.67	0.007	0.78	0.12	0.05	0.70	0.009	0.81	0.10	0.04	0.67
50%	0.003	0.70	0.14	0.10	0.64	0.008	0.70	0.19	0.08	0.65	0.006	0.73	0.11	0.05	0.68	0.008	0.80	0.09	0.04	0.66
m%	FCF predicts $af(u \rightarrow a)$										SCF predicts $af(u \rightarrow a)$									
	On Foursquare					On Gowalla					On Foursquare					On Gowalla				
	D^{hl}_{af}	D^{hl}_{pp}	$\overline{P^h}$	$\overline{P^l}$	$P^{h>l}$	D^{hl}_{af}	D^{hl}_{pp}	$\overline{P^h}$	$\overline{P^l}$	$P^{h>l}$	D^{hl}_{af}	D^{hl}_{pp}	$\overline{P^h}$	$\overline{P^l}$	$P^{h>l}$	D^{hl}_{af}	D^{hl}_{pp}	$\overline{P^h}$	$\overline{P^l}$	$P^{h>l}$
10%	0.003	0.98	0.13	0.02	0.94	0.003	0.98	0.08	0.02	0.85	0.025	0.89	0.20	0.03	0.90	0.008	0.67	0.17	0.07	0.71
20%	0.002	0.97	0.12	0.03	0.91	0.002	0.98	0.07	0.03	0.83	0.024	0.86	0.18	0.04	0.87	0.007	0.64	0.16	0.06	0.70
30%	0.001	0.97	0.11	0.03	0.89	0.003	0.97	0.07	0.03	0.81	0.022	0.84	0.17	0.05	0.85	0.007	0.62	0.15	0.06	0.69
40%	0.001	0.95	0.11	0.04	0.87	0.002	0.96	0.07	0.03	0.79	0.021	0.82	0.16	0.05	0.83	0.006	0.60	0.14	0.06	0.68
50%	0.001	0.93	0.10	0.04	0.85	0.001	0.95	0.07	0.04	0.77	0.019	0.79	0.16	0.07	0.80	0.006	0.58	0.14	0.06	0.67
m%	GCF predicts $af(u \rightarrow a)$										CCF predicts $af(u \rightarrow a)$									
	On Foursquare					On Gowalla					On Foursquare					On Gowalla				
	D^{hl}_{af}	D^{hl}_{pp}	$\overline{P^h}$	$\overline{P^l}$	$P^{h>l}$	D^{hl}_{af}	D^{hl}_{pp}	$\overline{P^h}$	$\overline{P^l}$	$P^{h>l}$	D^{hl}_{af}	D^{hl}_{pp}	$\overline{P^h}$	$\overline{P^l}$	$P^{h>l}$	D^{hl}_{af}	D^{hl}_{pp}	$\overline{P^h}$	$\overline{P^l}$	$P^{h>l}$
10%	0.011	0.91	0.21	0.04	0.87	0.002	0.73	0.16	0.09	0.65	0.002	0.90	0.14	0.03	0.89	0.002	0.85	0.23	0.004	0.88
20%	0.010	0.88	0.19	0.05	0.84	0.002	0.72	0.16	0.08	0.65	0.002	0.86	0.13	0.03	0.86	0.004	0.84	0.20	0.003	0.80
30%	0.010	0.86	0.18	0.06	0.83	0.002	0.71	0.14	0.08	0.63	0.001	0.84	0.12	0.04	0.83	0.003	0.80	0.18	0.003	0.74
40%	0.009	0.84	0.17	0.06	0.80	0.002	0.70	0.13	0.08	0.62	0.001	0.81	0.11	0.04	0.81	0.003	0.77	0.13	0.004	0.66
50%	0.008	0.81	0.17	0.07	0.77	0.001	0.67	0.13	0.09	0.61	0.001	0.78	0.11	0.05	0.79	0.003	0.75	0.09	0.004	0.60

probabilities, a large D^{hl}_{pp} indicates that the users have higher partner probabilities to activities in A^h (than in A^l).

— Under the above scenario, we can see that $\overline{P^h}$ is much higher than $\overline{P^l}$, which supports that users tend to attend activities in A^h (i.e., those having higher partner probabilities). To measure the significance of the difference between P^h and P^l, we use the t-test (setting the significance level α to 0.05) for testing paired difference hypothesis. The null hypothesis is "the difference between P^h and P^l samples is not significant." After testing, we found that that the null hypothesis is rejected under all cases, which indicates that the differences between $\overline{P^h}$ and $\overline{P^l}$ are statistically significant. Moreover, we observe that $P^{h>l}$ is much larger than 0.5 in all cases, which indicates the P^h is always larger than P^l.

All the above results verify that, when users have similar attendance preferences to some activities, they tend to attend the ones that give them higher partner probabilities.

4.4.2 Comparison Results on Recommending Activities. We now verify the effectiveness of our partner-aware activity recommendation strategies on improving conventional recommenders at the task of recommending social activities. We compare the activity recommendation performances of conventional recommenders to their partner-aware counterparts (i.e., UCF vs. PAUCF, ICF vs. PAICF, FCF vs. PAFCF, SCF vs. PASCF, GCF vs. PAGCF, and CCF vs. PACCF). The number k of recommended items varies from 1 to 10 and the parameter ρ in our algorithm ranges from 2 to ∞. Here $\rho = \infty$ indicates that we use the partner probability to adjust the order of the whole list of activities.

Table 5.

k	(%)	On Foursquare				On Gowalla				On Foursquare				On Gowalla			
		UCF	PAUCF			UCF	PAUCF			ICF	PAICF			ICF	PAICF		
			ρ=2	10	∞		ρ=2	10	∞		ρ=2	10	∞		ρ=2	10	∞
1	Precision	4.88	5.17	5.36	5.65	5.37	5.98	6.58	5.81	1.78	1.98	2.67	3.21	3.11	3.11	3.65	4.87
	Recall	2.07	2.19	2.28	2.40	1.10	1.23	1.35	1.19	0.59	0.66	0.89	1.08	0.46	0.46	0.54	0.72
2	Precision	4.45	4.83	5.22	5.07	4.79	5.44	6.12	5.10	1.64	2.15	2.53	2.77	2.82	3.14	3.38	4.56
	Recall	3.78	4.11	4.43	4.31	1.97	2.24	2.52	2.10	1.10	1.44	1.70	1.86	0.83	0.93	1.00	1.35
5	Precision	3.79	4.21	4.11	4.02	3.78	4.76	4.72	4.34	1.36	1.76	2.02	2.24	2.51	2.74	2.83	3.66
	Recall	8.06	8.96	8.75	8.55	3.90	4.91	4.87	4.48	2.29	2.96	3.40	3.77	1.86	2.03	2.09	2.71
10	Precision	3.00	3.44	3.28	3.24	3.22	4.30	3.93	3.63	1.21	1.44	1.78	1.79	2.15	2.46	2.45	2.90
	Recall	12.7	14.6	13.9	13.8	6.63	8.86	8.11	7.48	4.06	4.85	6.0	6.02	3.18	3.64	3.63	4.29

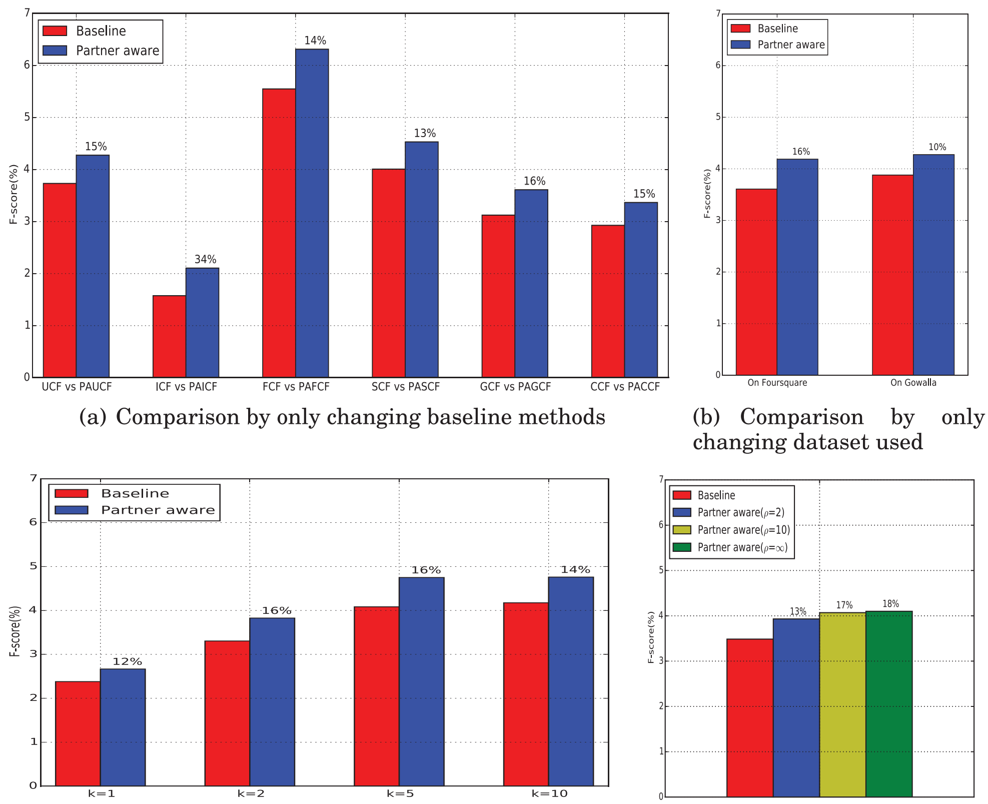
k	(%)	On Foursquare				On Gowalla				On Foursquare				On Gowalla			
		FCF	PAFCF			FCF	PAFCF			SCF	PASCF			SCF	PASCF		
			ρ=2	10	∞		ρ=2	10	∞		ρ=2	10	∞		ρ=2	10	∞
1	Precision	6.60	6.60	7.34	7.78	8.04	8.65	8.51	8.12	4.74	5.41	5.60	5.50	6.91	7.63	7.41	7.21
	Recall	4.33	4.33	4.82	5.11	2.43	2.61	2.57	2.45	2.19	2.5	2.58	2.54	1.42	1.57	1.52	1.48
2	Precision	5.43	6.31	6.09	6.46	7.06	7.60	7.41	7.41	4.13	4.74	4.98	5.08	5.94	6.59	6.79	6.35
	Recall	7.13	8.29	8.00	8.48	4.27	4.60	4.48	4.48	3.81	4.38	4.60	4.69	2.44	2.71	2.79	2.61
5	Precision	3.81	4.22	4.43	4.75	5.43	5.92	6.12	5.99	3.43	3.57	3.74	3.68	4.52	5.67	5.56	5.31
	Recall	12.5	13.8	14.5	15.6	8.20	8.94	9.24	9.05	7.93	8.24	8.64	8.50	4.65	5.83	5.72	5.46
10	Precision	2.86	3.32	3.95	3.96	4.35	4.77	5.04	5.06	2.67	2.84	2.91	2.87	4.06	4.83	4.56	4.43
	Recall	18.8	21.7	25.9	26.0	13.1	14.3	15.1	15.2	12.3	13.1	13.4	13.2	8.34	9.92	9.37	9.09

k	(%)	On Foursquare				On Gowalla				On Foursquare				On Gowalla			
		GCF	PAGCF			GCF	PAGCF			CCF	PACCF			CCF	PACCF		
			ρ=2	10	∞		ρ=2	10	∞		ρ=2	10	∞		ρ=2	10	∞
1	Precision	3.89	4.29	4.37	4.69	5.01	5.37	5.89	5.64	3.62	3.84	3.62	4.06	4.64	4.82	5.54	5.73
	Recall	1.74	1.92	1.96	2.10	1.05	1.12	1.23	1.18	1.48	1.57	1.48	1.65	0.90	0.93	1.07	1.11
2	Precision	3.42	3.46	3.77	3.81	4.27	5.24	5.40	5.16	3.59	3.84	3.84	4.24	3.97	4.82	5.04	5.16
	Recall	3.06	3.10	3.38	3.42	1.79	2.20	2.26	2.16	2.93	3.14	3.14	3.46	1.54	1.87	1.95	2.00
5	Precision	2.56	2.92	3.16	3.03	3.56	4.30	4.31	4.02	2.83	2.96	3.30	3.28	3.42	3.82	4.05	4.19
	Recall	5.74	6.56	7.09	6.81	3.73	4.51	4.52	4.22	5.77	6.04	6.75	6.69	3.31	3.70	3.92	4.07
10	Precision	2.22	2.59	2.64	2.64	3.13	3.64	3.5	3.37	2.10	2.40	2.63	2.53	2.88	3.13	3.19	3.39
	Recall	9.98	11.6	11.8	11.8	6.56	7.64	7.33	7.07	8.59	9.83	10.7	10.3	5.58	6.07	6.20	6.58

The precisions and recalls of all tested methods for different values of k and ρ are listed in Table 5. Figure 9 summarizes the results in a more comprehensive way, by grouping the experimental instances by baseline method, used dataset, ρ and k. The figure also explicitly shows the improvement that our partner-aware strategy achieves in each case. The F-score is defined as

$$F\text{-}score = \frac{2 \times (Precision \times Recall)}{Precision + Recall}. \tag{24}$$

From the results, we can see that the partner-aware methods significantly outperform conventional methods. Specifically, from Figure 9(a) we can see that the improvement is different when a different baseline method is used. The partner-aware strategy offers the largest improvement (34%) to ICF. Among the baseline approaches, FCF performs best, while PAFCF offers clear improvement (14%) to it. Figure 9(b) and 9(c) show that partner-aware recommendation offers a



(a) Comparison by only changing baseline methods

(b) Comparison by only changing dataset used

(c) Comparison by only changing values of k

(d) Comparison by only changing values of ρ

Fig. 9. Performance comparison for recommending activities.

stable improvement regardless the data used and the value of k. Finally, Figure 9(d) suggests the choice of ρ when applying partner-aware activity recommendation. Naturally, the improvement increases with ρ, since more items are considered as candidates. There is a jump in the improvement from $\rho = 2$ to $\rho = 10$, indicating that including more candidates in the partner-aware ranking module improves the quality of the selected results. On the other hand, the improvement from $\rho = 10$ to $\rho = \infty$ is negligible. The reason is that the bottom-ranked activity candidates by the baseline recommenders have very low attendance preferences; although the partner factor may increase their final scores, they cannot reach the top k list. In summary, a very large value for ρ does not bring a big improvement in the quality of the recommended results. On the other hand, if ρ is very large, it brings a large computation burden to the recommender, because it proportionally increases the number of partner probabilities to be computed. Each calculation of partner probability requires computing together preferences to all partner candidates. Thus, we suggest using a medium value for ρ (e.g., $\rho = 10$) in practice.

5 RELATED WORK

5.1 Recommender Systems

Previous research on recommender systems can be classified into two categories. Research in the first class focuses on the design of models that improve the accuracy of general-purpose (i.e., classic) recommendation tasks (e.g., recommendation of books or movies). Another category, which has gained popularity in recent years, includes efforts that discover interesting applications of recommender systems and extend base recommenders to domain-specific tasks (He et al. 2010; Wu et al. 2013). Our work is in both of these directions. We study a new recommendation problem: recommend partners for the activity items suggested to a user. Moreover, we also improve activity recommendation by taking the probability that the user can find activity partners into consideration.

5.2 Friend Recommendation

Social networking services (e.g., MySpace, Facebook) enable people to become friends with each other. The effectiveness of these services depends on the quality of the social connections between their members. As a result, friend recommendation (Hannon et al. 2010) became a popular research topic, assisting social networks to improve their service. For example, Facebook has launched the tool "People You May Know" to recommend friends based on social proximity. This tool recommends "friends of friends": if A knows B and B knows C, then Facebook tells A "You May Know C."

Commonly to friend recommendation, the recommended object in our problem is also a user. However, the tasks of friend recommendation and activity-partner recommendation are very different. Friend recommendation systems predict user-user relationships (i.e., friendships) (Chen et al. 2009; Zheng et al. 2011), while our work explores (user, item)-user relationship, i.e., together preference from a (user, activity item) to an activity partner. Friend recommendation estimates the likelihood that two non-friends will become friends in the future. On the other hand, in a real-world application of activity-partner recommendation, the candidates of activity partners are limited to users who are already the target user's friends. Actually, the *SCAPR* method, which employs the social closeness between users to recommend activity partners, uses the idea of transferring social connections in friend recommendation research into our partner recommendation problem.

5.3 Group Recommendation

Group recommendation (Gorla et al. 2013; Yuan et al. 2014) explores the preference of a group of users to individual items. Currently, many services (e.g., Movielens) allow the creation of groups that consist of several users. Then a typical objective of group recommendation is to aggregate the preferences of group members to find relevant items for groups. The problem of activity-partner recommendation is different from the problem of group recommendation. Most works in group recommendation aim at selecting items for fixed groups (Gorla et al. 2013; Yuan et al. 2014; Jameson and Smyth 2007; Amer-Yahia et al. 2009; Senot et al. 2010), while activity-partner recommendation strives to find users as activity partners having as fixed variables a target user and an activity item (recommended by any activity-item recommendation system).

However, we can still adapt group recommendation approaches to achieve activity-partner recommendation. An idea is to regard the target user and each activity-partner candidate as a simulated group. Then, we can select the candidates that participate in the groups with the highest group preference to the given activity item. Actually, since in the problem of recommending activity partners the target user is fixed, the preference from the target user to the activity item is also fixed. Therefore, the above idea is the same as recommending the partner candidates by their preferences to the activity item (i.e., our *ALAPR* method).

Some works in group-recommendation research generate simulated user groups for their experiments if they lack ground truth of groups. For this purpose, they employ similarity in preferences (Purushotham et al. 2014; Ntoutsi et al. 2012), or social-demographic information (Yu et al. 2006; Lieberman et al. 1999; Crossen et al. 2002). Note that these works always generate groups before recommending items. Therefore, they focus on user-user relationships, while our together preference is (user, item)-user relationship. Actually, our SIAPR and SCAPR can be regarded as employing group generation strategies with interest similarity and social links for recommending activity partners.

5.4 Location Recommendation on LBSN

A LBSN adds locations to an existing social network so that people in the network can share location-embedded information. Location recommendation is an important feature of social network applications and location-based services (Yin et al. 2014; Wang et al. 2016). While classic CF algorithms can be adjusted to the problem of recommending new locations to users (Zheng et al. 2009; Bao et al. 2015), by taking into account previous user check-ins, significant information like the distance of the proposed location to the user neighborhood or the social interaction between the users are ignored. Recent methods exploit geographical and social information for generating recommendations (Ye et al. 2011; Scellato et al. 2011; Chen et al. 2012; Ye et al. 2012; Liu et al. 2013).

Our work is designed for social-activity items, which have differences compared to locations. In other words, we consider items such as movie tickets and restaurant discount coupons related to activities in which people like to participate with their folks. As we analyzed in the experiments, many locations (selected by social-related keywords) can be naturally regarded as social activities. Therefore, our work can be used for improving recommendations for activity-related locations. The first part of our work (activity-partner recommendation) focuses on recommending peer partners, while LBSN websites recommend activity-related locations to single users. The second part (partner-aware activity recommendation) considers the partner factor when recommending activity-related locations. Our experiments were conducted on LBSN datasets and verify the assumption that people tend to go to activity-related locations for which they can find suitable partners. Our results also show that our work improves upon several commonly used methods for recommending activity-related locations.

6 CONCLUSION AND FUTURE WORK

In this article, we first propose and study the problem of recommending activity partners to Web users for activity items suggested to them. We explore how to take advantage of different types of data and relationships, including the attendance preference by users to activities, the social context of users, and the past together preference knowledge in order to solve our activity-partner recommendation problem.

Secondly, we study the use of partner factor for improving activity recommendation. We use the together preference estimator of our activity-partner recommenders to predict partner probability, which indicates whether a user can find partners for attending an activity together. Then, by integrating partner factor into conventional activity recommenders that consider users' interests on activities, we propose partner-aware activity recommendation. We expect that users not only are interested in the recommended activities but also prefer to find partners to attend them together.

In our experiments, based on a questionnaire, we verify that real users have great interest in activity-partner recommendation. Then, we utilize datasets from location-based social networks to simulate an activity recommendation scenario by identifying locations related to social activities and extracting ground truth of activity partners. We perform experiments to analyze the strengths and weaknesses of recommending activity partners. We also compare the performance of activity recommenders with and without considering the partner factor. The results verify the effectiveness of our partner-aware framework for improving the quality of recommended social activity items.

The most important subject in our future work is to realize activity-partner recommendation in real-world systems. We expect that social-network-based companies (e.g., Facebook, Tencent QQ) and activity-related product websites (e.g., Groupon) would have great interest in employing the idea of recommending products and partners together. Then, this new recommendation problem can attract attention and be studied further. We anticipate that implementing activity-partner recommendation will bring people's friendship from the virtual world into their real lives. Social networks enable users to communicate with each other on the Web. However, a user may have people in her list of friends who live in the same city, but with whom she barely socializes in her real life. Our activity-partner recommender can help such users to get in real contact with such friends. This way, we hope to increase the opportunities for people to meet their virtual friends by participating in real-life activities with them.

REFERENCES

L. A. Adamic and E. Adar. 2003. Friends and neighbors on the web. *Social Networks* 25, 3 (2003), 211–230.

S. Amer-Yahia, S. B. Roy, A. Chawlat, G. Das, and C. Yu. 2009. Group recommendation: Semantics and efficiency. *Proceedings of the VLDB Endowment* (2009), 754–765.

A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec. 2012. Effects of user similarity in social media. In *Proceedings of the ACM International Conference on Web Search and Data Mining*. ACM, 703–712.

J. Bao, Y. Zheng, and M. F. Mokbel. 2012. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the International Conference on Advances in Geographic Information Systems*. ACM, 199–208.

J. Bao, Y. Zheng, D. Wilkie, and M. Mokbel. 2015. Recommendations in location-based social networks: A survey. *GeoInformatica* 19, 3 (2015), 525–565.

C. Chen, Y. Haiqin, K. Irwin, and L. Michael. 2012. Fused matrix factorization with geographical and social influence in location-based social networks. In *AAAI Conference on Artificial Intelligence*.

J. Chen, W. Geyer, C. Dugan, M. Muller, and I. Guy. 2009. Make new friends, but keep the old: Recommending people on social networking sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 201–210.

E. Cho, S. A Myers, and J. Leskovec. 2011. Friendship and mobility: User movement in location-based social networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1082–1090.

C. Y. Chow, J. Bao, and M. F. Mokbel. 2010. Towards location-based social networking services. In *Proceedings of the ACM SIGSPATIAL International Workshop on Location Based Social Networks*. ACM, 31–38.

A. Crossen, J. Budzik, and K. J. Hammond. 2002. Flytrap: Intelligent group music recommendation. In *Proceedings of the 7th International Conference on Intelligent User Interfaces*. ACM, 184–185.

H. Gao, J. Tang, and H. Liu. 2012. gSCorr: Modeling geo-social correlations for new check-ins on location-based social networks. In *Proceedings of the ACM International Conference on Information and Knowledge Management*. ACM, 1582–1586.

J. Gorla, N. Lathia, S. Robertson, and J. Wang. 2013. Probabilistic group recommendation via information matching. In *Proceedings of the International Conference on World Wide Web*. 495–504.

A. Gunawardana and G. Shani. 2009. A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research* 10 (2009), 2935–2962.

J. Hannon, M. Bennett, and B. Smyth. 2010. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings of the ACM Conference on Recommender Systems*. 199–206.

Q. He, J. Pei, D. Kifer, P. Mitra, and L. Giles. 2010. Context-aware citation recommendation. In *Proceedings of the International Conference on World Wide Web*. 421–430.

T. Horozov, N. Narasimhan, and V. Vasudevan. 2006. Using location for personalized POI recommendations in mobile environments. In *Proceedings of the International Symposium on Applications and the Internet*. IEEE.

A. Jameson and B. Smyth. 2007. Recommendation to groups. In *The Adaptive Web*. Springer, 596–627.

Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel. 2012. Lars: A location-aware recommender system. In *International Conference on Data Engineering*. IEEE, 450–461.

Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W. Ma. 2008. Mining user similarity based on location history. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 34.

H. Lieberman, N. Van Dyke, and A. Vivacqua. 1999. Let's browse: A collaborative browsing agent. *Knowledge-Based Systems* 12, 8 (1999), 427–431.

B. Liu, Y. Fu, Z. Yao, and H. Xiong. 2013. Learning geographical preferences for point-of-interest recommendation. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1043–1051.

A. Noulas, S. Scellato, C. Mascolo, and M. Pontil. 2011. An empirical study of geographic user activity patterns in foursquare.*International Conference on Weblogs and Social Media* 11 (2011), 70–573.

E. Ntoutsi, K. Stefanidis, K. Nørvåg, and H. P. Kriegel. 2012. Fast group recommendations by applying user clustering. In *Conceptual Modeling*. Springer, 126–140.

S. Purushotham, C. J. Kuo, J. Shahabdeen, and L. Nachman. 2014. Collaborative group-activity recommendation in location-based social networks. In *Proceedings of the ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information.* ACM, 8–15.

R. W. Root and S. Draper. 1983. Questionnaires as a software evaluation tool. In *Proceedings of the Conference on Human Factors in Computing Systems.* 83–87.

B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the International Conference on World Wide Web*. 285–295.

S. Scellato, A. Noulas, and C. Mascolo. 2011. Exploiting place features in link prediction on location-based social networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1046–1054.

J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. 2007. Collaborative filtering recommender systems. In *The Adaptive Web*. 291–324.

C. Senot, D. Kostadinov, M. Bouzid, J. Picault, A. Aghasaryan, and C. Bernier. 2010. Analysis of strategies for building group profiles. In *Proceedings of the International Conference on User Modeling, Adaptation, and Personalization.* Springer, 40–51.

W. Tu, D. Cheung, N. Mamoulis, M. Yang, and Z. Lu. 2015. Activity-partner recommendation. In *Advances in Knowledge Discovery and Data Mining*. Springer, 591–604.

H. Wang, M. Terrovitis, and N. Mamoulis. 2013. Location recommendation in location-based social networks using user check-in data. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.* ACM, 374–383.

W. Wang, H. Yin, L. Chen, Y. Sun, S. Sadiq, and X. Zhou. 2015. Geo-SAGE: A geographical sparse additive generative model for spatial item recommendation. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1255–1264.

W. Wang, H. Yin, S. Sadiq, L. Chen, M. Xie, and X. Zhou. 2016. SPORE: A sequential personalized spatial item recommender system. In *International Conference on Data Engineering (ICDE)*. IEEE.

S. Wu, J. Sun, and J. Tang. 2013. Patent partner recommendation in enterprise social networks. In *Proceedings of the ACM International Conference on Web Search and Data Mining*. 43–52.

X. Xiao, Y. Zheng, Q. Luo, and X. Xie. 2010. Finding similar users using category-based location history. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 442–445.

J. Ye, Z. Zhu, and H. Cheng. 2013. What's your next move: User activity prediction in location-based social networks. In *Proceedings of the SIAM International Conference on Data Mining*.

M. Ye, X. Liu, and W. C. Lee. 2012. Exploring social influence for recommendation: A generative model approach. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 671–680.

M. Ye, P. Yin, and W. C. Lee. 2010. Location recommendation for location-based social networks. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.* ACM, 458–461.

M. Ye, P. Yin, W.C. Lee, and D. L. Lee. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval.* 325–334.

H. Yin, B. Cui, Y. Sun, Z. Hu, and L. Chen. 2014. LCARS: A spatial item recommender system. *ACM Transactions on Information Systems* 32, 3 (2014), 11.

H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen. 2013. LCARS: A location-content-aware recommender system. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 221–229.

H. Yin, X. Zhou, Y. Shao, H. Wang, and S. Sadiq. 2015. Joint modeling of user check-in behaviors for point-of-interest recommendation. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management.* ACM, 1631–1640.

J. Ying, H. Lu, W. Kuo, and V. Tseng. 2012. Urban point-of-interest recommendation by mining user check-in behaviors. In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*. ACM, 63–70.

Z. Yu, X. Zhou, Y. Hao, and J. Gu. 2006. TV program recommendation for multiple viewers based on user profile merging. *User Modeling and User-adapted Interaction* 16, 1 (2006), 63–82.

Q. Yuan, G. Cong, and C. Y. Lin. 2014. COM: A generative model for group recommendation. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 163–172.

W. Zhang and J. Wang. 2015. A collective Bayesian Poisson factorization model for cold-start local event recommendation. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 1455–1464.

Y. Zheng and X. Xie. 2010. Learning location correlation from GPS trajectories. In *International Conference on Mobile Data Management*. IEEE, 27–32.

Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.-Y. Ma. 2011. Recommending friends and locations based on individual location history. *ACM Transactions on the Web* 5, 1 (2011), 5.

Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. 2009. Mining correlation between locations using human location history. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 472–475.

Received December 2015; revised May 2017; accepted July 2017