Activity-Partner Recommendation

Wenting Tu, David W. Cheung, Nikos Mamoulis, Min Yang, and Ziyu Lu

Department of Computer Science, The University of Hong Kong
Pokfulam, Hong Kong
{wttu, dcheung, nikos, myang, zylu}@cs.hku.hk

Abstract. In many activities, such as watching movies or having dinner, people prefer to find partners before participation. Therefore, when recommending activity items (e.g., movie tickets) to users, it makes sense to also recommend suitable activity partners. This way, (i) the users save time for finding activity partners, (ii) the effectiveness of the item recommendation is increased (users may prefer activity items more if they can find suitable activity partners), (iii) recommender systems become more interesting and enkindle users' social enthusiasm. In this paper, we identify the usefulness of suggesting activity partners together with items in recommender systems. In addition, we propose and compare several methods for activity-partner recommendation. Our study includes experiments that test the practical value of activity-partner recommendation and evaluate the effectiveness of all suggested methods as well as some alternative strategies.

1 Introduction

In real-world recommendation applications, many items are related to activities that people like to participate with their folks. For example, items such as online game invitations, movie tickets, dinner discounts are related to social activities (playing games, 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(a).

Previous work on recommending activity items typically focused on improving the precision, recall, or diversity of recommended items [1]. In this paper, we follow a totally new direction: as Figure 1 shows, instead of recommending only activity items to users, we combine the activity-item sale platform and social network platform to make the activity-item sales benefit from also recommending *activity partners* for the suggested items. Our rationale is that, for activities in which people like to participate with their folks, if a system recommends a related item alone, the user may give up attending the activity (i.e., reject the item) if s/he cannot immediately think of someone to invite to attend the activity together. The Figure 1(c) illustrates the effectiveness of recommending activity partners via an example. The recommended product "tickets of Bruno Mars' concert" is an activity item and the corresponding activity can be described as "watching Bruno Mars' concert". Imagine that you have some interest in Bruno Mars' show; however, when you see the recommendation message, it may be hard for you to think of suitable partners for watching the show together. This could be a good reason for you to give up attending this activity since you don't feel like going to 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. Based on this example, we designed a simple questionnaire to collect feedback from real web-users on the potential effectiveness of recommending activity partners. 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, improving therefore the revenue of the involved businesses.

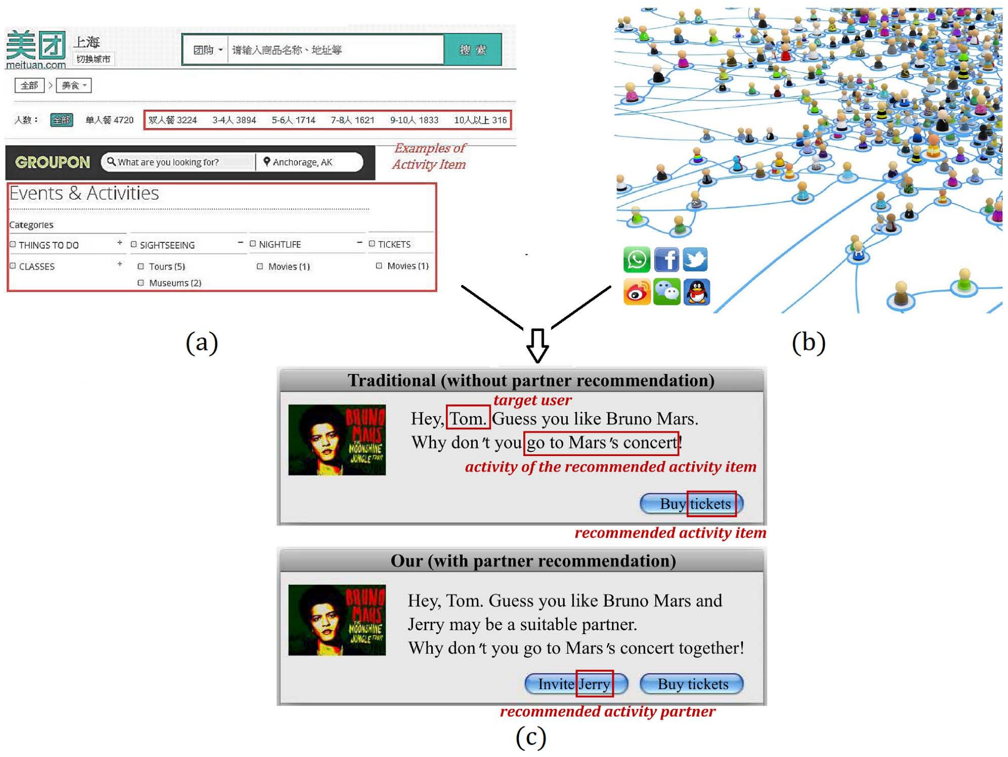

Examples of Activity Item
(a)
(b)
Traditional (without partner recommendation)
target user
Hey, Tom. Guess you like Bruno Mars.
Why don't you go to Mars's concert!
activity of the recommended activity item
Buy tickets
recommended activity item
Our (with partner recommendation)
Hey, Tom. Guess you like Bruno Mars and
Jerry may be a suitable partner.
Why don't you go to Mars's concert together!
Invite Jerry
Buy tickets
recommended activity partner
(c)

Fig. 1. Our recommendation service.

As discussed above, recommending activity partners is likely to improve the success rate of activity item recommendations. On the other hand, to the best of our knowledge, there are no previous studies or applications of this idea in the research community or the industry, respectively. This motivates us to investigate methods for activity-partner recommendation. We firstly explore how attendance preference and social context can be used to recommend activity partners. Then, we propose a method that analyzes the historical records of user preferences on activity partners to predict future activity partners. This is a reasonable methodology, since the past user preferences on activity partners would be available after the activity-partner recommendation system has been set up and used to collect training data.

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

– We bring in the idea of recommending suitable activity partners, in order to improve the effectiveness of activity item recommendation. A survey was conducted to confirm that real users favor 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 using user-item preferences and the social context of users. Since such data are commonly tracked in current recommendation systems, our results can directly be embedded into an existing recommender system to turn it into an activity-partner recommender.
– We also propose a methodology for recommending activity partners based on past partner knowledge of users. This method extends conventional collaborative filtering techniques to make them more suitable for our problem.
– We conduct an experimental evaluation based on real data that tests the effectiveness of all proposed methods in recommending activity partners.

The remainder of this paper is organized as follows. Section 2 formulates our problem. Section 3 describes our methods for activity-partner recommendation. Section 4 includes our experiments. Section 5 discusses related work. Finally, Section 6 concludes with a discussion about future work.

2 Problem Formulation

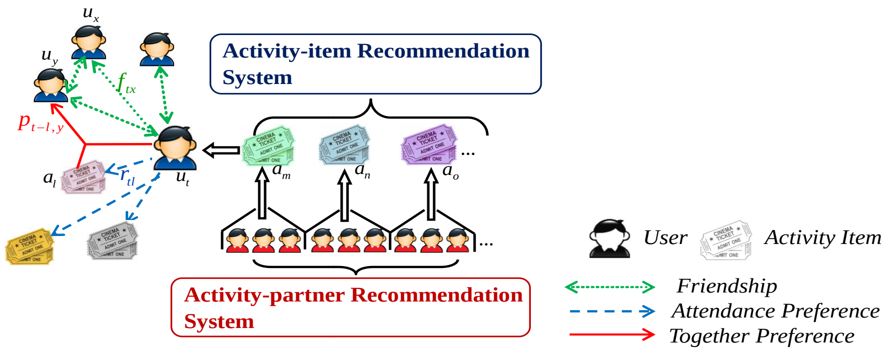


Fig. 2. Illustration of activity-partner recommendation.

As Figure 2 shows, typically there are two types of objects (i.e., *user* and *item*) in recommendation systems for activity items. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_{n_u}\}$ be the set of users and $\mathcal{A} = \{a_1, a_2, \ldots, a_{n_a}\}$ be the set of activity items. Two common types of relationships exist among these entities. First, 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. Second, users may indicate their preference to activity items. 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 use $pf(u_t \rightarrow a_l)$ to denote how much u_t prefers a_l. $pf(u_t \rightarrow a_l)$ can take value from a range (e.g., 1 to 5) or it can be binary number (i.e., $pf(u_t \rightarrow a_l) = 1$ means that u_t likes a_l). 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(c), this indicates that Tom prefers to attend activity "Bruno Mars' show" together with Jerry. The together preference relates a user and an activity item $[u_t, a_l]$, i.e., a *ua-pair*, to another user u_x. For example, the fact that Tom prefers the tickets of Bruno Mars' concert creates pair [Tom, tickets of Bruno Mars' concert]; if Tom likes Jerry to join him to the concert, then there is a relationship between [Tom, tickets of

Bruno Mars' concert] and Jerry. We use $pf([u_t, a_l] \rightarrow u_x)$ 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 p([Tom, tickets of Bruno Mars' concert]→ Jerry) to 1 if Tom clicks the "Invite Jerry" button or to 0 if Tom does not click the button.

As Figure 2 shows, the objective of our work is, for each activity item recommended by an activity-item recommendation system, to predict the users' together preference on the activity item. With the above notation, our problem can be stated as follows. Given a target user u_t and an activity item a_l (recommended by some activity-item recommender), use any known friendship, attendance preference, and together preference relationships to estimate $p([u_t, a_l] \rightarrow u_c)$, where u_c is any candidate activity partner. Then, rank the partner candidates u_c by their $pf([u_t, a_l] \rightarrow u_c)$ values and extract the top-k candidates as the recommended activity partners.

3 Activity-Partner Recommendation

3.1 Utilizing Attendance Preference and Social Context

In this section, we first utilize attendance preference and social context to implement activity-partner recommendation.

Social-Closeness 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* [2] (commonly used owing to its low computational complexity) to model the social closeness $SC(u_t, u_c)$ between two users: Thus, one user-user relationship that may help predict together preference $pf([u_t, a_l] \rightarrow u_x)$ is the social closeness $SC(u_t, u_c)$ between u_t and u_x. Here we assume that people prefer to attend activities with users who are socially close to them. Therefore, we can predict together preference as follows:

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

where $\mathcal{F}^t$ ($\mathcal{F}^c$) is the friends set of u_t (u_c). In order to recommend activity partners to a target user u_t, we rank the activity-partner candidates 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).*

Similar-Interest Hypothesis. The similarity between user interests (homophilly) is commonly used in previous recommender systems. For recommending activity partners based on user homophilly, 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 to them. For example, we can measure the cosine similarity between user-profile vectors and use it to define *Similar-Interest based Activity-Partner Recommendation (SIAPR)*:

$$p([u_t, a_l] \rightarrow u_c) \propto SI(u_t, u_c) = Sim_{Cosine}(\overline{r_t}, \overline{r_c}) = \frac{\overline{r_t} \cdot \overline{r_c}}{||\overline{r_t}||||\overline{r_c}||}, \tag{2}$$

where the vectors r_t and r_c capture the interests (i.e., the sets of preferred items) of u_t and u_c, respectively.

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

$$pf([u_t, a_l] \to u_c) \propto AL(u_c, a_l) = pf(u_c \to a_l), \tag{3}$$

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 [5] (explained in detail in Section 3.2) to estimate the attendance preference.

3.2 Utilizing Training Together Preference

In this section, we propose an alternative method to the simple strategies introduced in Section 3.1. Our objective is to predict a user's together preference via his/her past together preference records. We first discuss about 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.

Extracting 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 the of Figure 1(c), users' clicking behavior on the invitation button is a indicator of activity-partner preference. One typical 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 pm and 8:15 pm on the same day, most probably had dinner together.

Using 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". Up to now, two main classes of recommendation approaches exist: collaborative [3] or content-based filtering [4]. Content-based filtering methods extract features from the items and recommend to users items with similar features to past items chosen by them. In our problem, the "items" to be recommended are activity partners, which lack a generalized definition of content. Therefore, collaborative filtering (CF) appears to be a more suitable approach for activity-partner recommendation. Therefore, we propose a method, called *Collaborative Filtering based Activity-Partner Recommendation (CFAPR)*, which appropriately extends the idea of CF methods to solve our problem.

Before presenting *CFAPR*, we first explain how user-based CF [5] works. Since it can be used for predicting the attendance preferences in our APR problem, here we take the process of accessing $pf(u_t \to a_l)$ of single user u_t on item a_l as an example. The first step of the approach is to calculate for each other user u_i the vector similarity between rating profiles of u_t and u_i (denoted as $\overline{r_t}$ and $\overline{r_i}$). For example, we can use the similar interests Equation (2) to calculate the similarity $S^u_{t,i}$ between u_t and u_i. The second step is as follows: if $S^u_{t,i}$ satisfies some condition (e.g., 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 the preference $p^A_{t,l}$ of user u_t to activity a_l, we aggregate (weighted sum) the (known) preferences $p^A_{i,l}$ to a_l of all users u_i in the neighborhood of u_t, as follows:

$$p^A_{t,l} \propto \frac{1}{\sum_{u_i \in N^t} S^u_{t,i}} \sum_{u_i \in N^t} S^u_{t,i} r_{i,l}, \tag{4}$$

where N^t denotes the set of u_t's neighbor users who have rated a_l.

Now, assume that we try to apply this conventional user-based CF approach to predict the together preference $pf([u_t, a_l] \to u_c)$. Similarly, we can regard each $[u_t, a_l]$ as a special user unit. We call such a "user" unit a *ua-pair*. First, we should try to find the neighborhood of ua-pair $[u_t, a_l]$. However, since a conventional activity-item recommender always recommends to users activity items they have not rated yet, there must not be any historical together preference of $[u_t, a_l]$. The above fact 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 [6] or matrix-factorization-based CF [7] 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 to define 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 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 next 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 $[u_t, a_m]$. Note that we can also use content similarity between the activity items if the activity item carry a rich description. 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 $p^T_{t,l,c}$ (i.e., $pf([u_t, a_l] \to u_c)$) by aggregating all together preferences $p^T_{t,m,c}$ (i.e. $pf([u_t, a_m] \to u_c)$) of $[u_t, a_m]$ $(m \neq l)$ on u_c as:

$$p^T_{t,l,c} \propto \frac{1}{\sum_{[u_t,a_m] \in \mathcal{N}^{t,l}} S^a_{l,m}} \sum_{[u_t,a_m] \in \mathcal{N}^{t,l}} S^a_{l,m} p^T_{t,m,c}, \tag{5}$$

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*. Note that the size of the candidate set $\mathcal{C}^{t,l}$ of activity partners is an important parameter, since the problem size is determined by it. For example, in our experiments, we restrict the candidate set of activity partners to include only the users who have a friendship connection with u_t to control the problem size and the cost of *CFAPR*. Studying the effect of alternative methods for restricting the candidate-set is an important direction of our future work.

Algorithm 1 *CFAPR*

```
Input: (i), $\mathcal{C}^{t,l}$: 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 $a_l \in \mathcal{A}^i$
    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 $u_c \in \mathcal{C}^{t,l}$ do Compute $pf([u_t, a_l] \to u_c)$ using Eq.(5)
Return $K$ users in $\mathcal{C}^{t,l}$ having the highest $K$ $pf([u_t, a_l] \to u_*)$ values.
```

4 Experiments

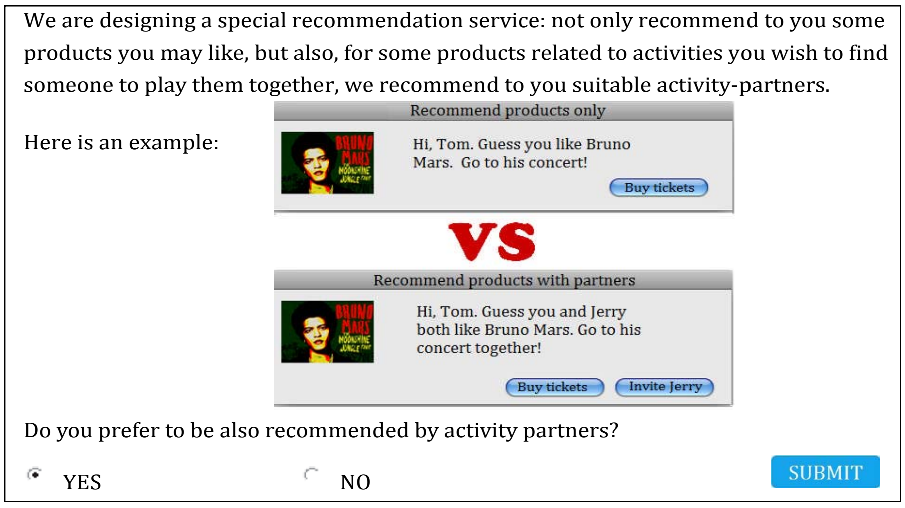
Section 4.1 demonstrates the meaningfulness of activity-partner recommendation via feedback collected from real web users. Section 4.2 evaluates the activity partner recommendation strategies described in Section 3.

4.1 Users' Favor of Activity-Partner Recommendation

To confirm the practical value of our work, we conducted an electronic survey 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, which has the format shown in Figure 3 asks people whether they prefer to be also recommended by activity partners and was released to public Chinese web-users from November 21, 2014. Until the submission of this work, 57 web-users (from various provinces of China) returned their answers to us. Although we did not get a lot of feedback (there were very few web users willing to fill-in the on-line questionnaire without a reward), we believe that the sample is big enough to reflect the opinion of typical web-users. Finally, *about 93% 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 Effectiveness of Activity-Partner Recommenders

Data. In our effectiveness evaluation, we used data from location-based social networks (LSBN) to simulate a real-world scenario for our problem. We regard locations in LSBN datasets as activity items. This is reasonable, since many activity items (e.g., tickets, dinner vouchers) refer to particular locations at particular time periods or moments. For example, location Han Dynasty (Chinese Restaurant, 4356 Main St, Philadelphia, PA 19127) can be regarded as activity-item "Coupon for eating Chinese food in Han Dynasty". Moreover, the activity partners with whom users attend (check-in) some activity items (location) can be inferred based on the check-in timestamps and the friendship relationships between users, as discussed in Section 3.2: if two users are friends and check-in at a same location at close timestamps (i.e., the time difference between their check-in timestamps is less than three hours), we regard that the two users attend the corresponding activity item (location) together and thus they are activity partners of



We are designing a special recommendation service: not only recommend to you some products you may like, but also, for some products related to activities you wish to find someone to play them together, we recommend to you suitable activity-partners.

Here is an example:

Recommend products only

Hi, Tom. Guess you like Bruno Mars. Go to his concert!

VS

Recommend products with partners

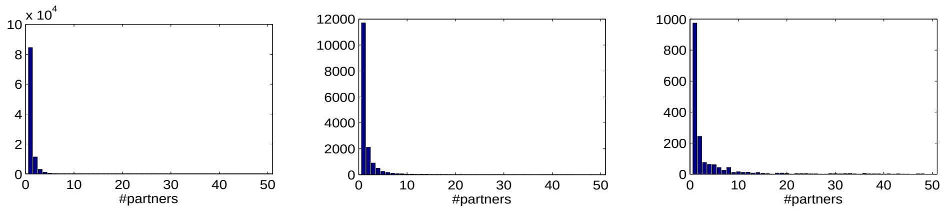
Hi, Tom. Guess you and Jerry both like Bruno Mars. Go to his concert together!

Do you prefer to be also recommended by activity partners?

[✓ YES] NO

Fig. 3. The questionnaire used to assess the favor of web-users to activity-partner recommendation.

each other with respect to the activity item. We used data[1] crawled from three popular LSBN websites: Gowalla (`gewalla.com`), Foursquare (`foursquare.com`) and Brightkite (`brightkite.com`). All these datasets have check-in timestamps and social links between users. Finally, we obtained 101400 (from Gowalla), 16220 (from Foursqure), and 1690 (from Brightkite) [user, activity] ua-pairs for testing (each such ua-pair is associated with at least one activity partners, e.g., [John, Han Dynasty]$\rightarrow$ {Jerry, Bella, Nicole$\cdots$})[2].



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

Fig. 4. The number of valid user-activity pairs (y-axis) having a given number of activity partners (x-axis).

Evaluation Measures. After extracting the testing ua-pairs and their corresponding activity-partner knowledge, we use the tested methods to recommend K activity partners for each valid user-activity pair. We denote the set of valid user-activity pairs as $\mathcal{V}$, the

[1] Released on `http://i.cs.hku.hk/~wttu/apr_project.html`

[2] We discard the ua-pairs without any extracted activity partners. For example, if we find that none of John's friends checked in Han Dynasty at a close timestamp to John's, we infer that there are no activity partners of the ua-pair [John, Han Dynasty]; thus, we do not use [John, Han Dynasty] as a ua-pair in our experiments.

real activity-partners of a testing ua-pair $[u_t, a_l]$ as $Pa^{real}(u_t, a_l)$, and the recommended partners to $[u_t, a_l]$ as $Pa^{rec}(u_t, a_l)$. We use the classic precision and recall measures to evaluate the performance of recommending activity partners.

$$Precision = \frac{\sum_{(u_t,a_l)\in\mathcal{V}} |Pa^{rec}(u_t, a_l) \bigcap Pa^{real}(u_t, a_l)|}{\sum_{(u_t,a_l)\in\mathcal{V}} |Pa^{rec}(u_t, a_l)|}, \tag{6}$$

$$Recall = \frac{\sum_{(u_t,a_l)\in\mathcal{V}} |Pa^{rec}(u_t, a_l) \bigcap Pa^{real}(u_t, a_l)|}{\sum_{(u_t,a_l)\in\mathcal{V}} |Pa^{real}(u_t, a_l)|}. \tag{7}$$

Competitors. Besides methods *SCAPR, SIAPR, ALAPR* (introduced in Section 3.1), and *CFAPR* (introduced in Section 3.2), we include in the evaluation an additional strategy, which also employs together preference training. This method is called *Popular-Partner based APR (PPAPR)* and models the popularity of a activity partner candidate by the times s/he is preferred as an activity partner *by the target user only.* PPAPR is based on a partner consistency hypothesis, while *CFAPR* assumes that partners are sensitive to activities. In PPAPR, the popularity of a partner candidate u_c for a target user u_t is defined as follows:

$$pf([u_t, a_l] \to u_c) \propto Pop(u_t, u_c) = |\mathcal{V}_c^t|, \tag{8}$$

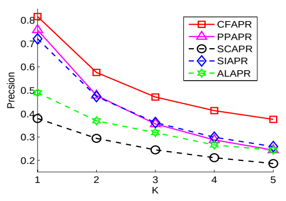
where $\mathcal{V}_c^t$ is the set of valid user-activity pairs of user u_t whose activity partners include u_c.

While evaluating all methods, for each testing user-activity pair (e.g. $[u_t, a_l]$), we set the candidate set of activity partners as the friends set of u_t. Moreover, we use all neighbor candidates in *CFAPR* as the ua-pair neighbors (set *neighbor_condition* in Algorithm 1 initially as TRUE) and use the Cosine similarity between activities' rated vectors. Besides, for implementing *ALAPR*, we used user-based CF as a basis with the neighbors of a user being the 100 most similar users to the target user.

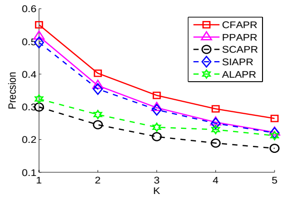
Results and Analysis. Before performing performance comparison, we analyze the number of activity partners users prefer when attending an activity. According to the check-in records of LBSN datasets, we found that most of (more than 95% in our experiments) the user-activity pairs have 1 to 5 activity partners. Therefore, we will test the performance of the methods introduced above on activity-partner recommendation when the size of recommendation list of activity partners is changing from 1 to 5. Figure 5 shows the results of all methods while K is varying from 1 to 5. Note that the precision of all methods falls and the recall increases as K increases, which indicates the predictions more close to the top are more accurate. When comparing performance of different methods, we can observe that:

– *CFAPR outperforms all other methods.*
  This indicates the suitability of *CFAPR* for activity-partner recommendation with training together-preference knowledge.
– *CFPAR outperforms PPAPR.*
  Both of *CFAPR* and *PPAPR* take use of past together preference. The difference between *CFAPR* and *PPAPR* is that *CFPAR* considers the influence of activity item together preferences. *CFAPR* assumes that the together preferences of a user on similar activity items are similar. The fact that *CFAPR* outperforms *PPAPR* has verified this assumption.

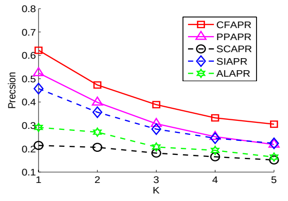
– *CFPAR and PPAPR outperform SIAPR, SCAPR, ALAPR.*
  In general, the methods which use past together preferences (i.e., *CFAPR* and *PPAPR*) of the target user perform better than methods which ignore this parameter (i.e., *SIAPR*, *SCAPR*, *ALAPR*). This fact shows that past together preferences play an important role in predicting activity partners.
– *SIAPR outperforms SCAPR, ALAPR.*
  SIAPR, *ALAPR*, *SCAPR* are three methods which uses information commonly seen in many current 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 raining together-preference knowledge, *SIAPR* is a good choice to start up a activity-partner recommendation system.



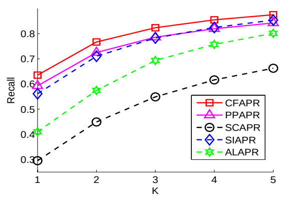
(a) Precision (Gowalla)



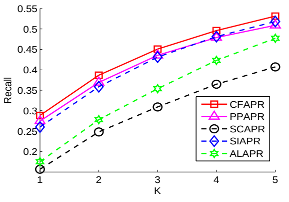
(b) Precision (Foursquare)



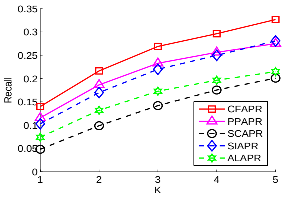
(c) Precision (Brightkite)



(d) Recall (Gowalla)



(e) Recall (Foursquare)



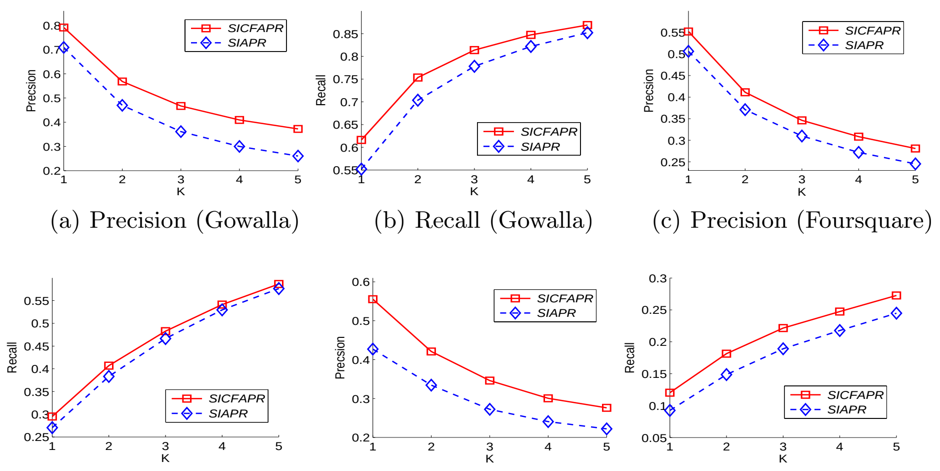
(f) Recall (Brightkite)

Fig. 5. Performance comparison of methods *CFAPR*, *PPFAPR*, *SCAPR*, *SIAPR* and *ALAPR*.

Note that results in Figure 5 are on warm-start users; prior knowledge of together preferences is a requirement for methods such as *CFAPR* and *PPAPR*. The results show that *CFAPR* performs best for this set of users. In the case, where there are users with no past together preferences, we propose a hybrid strategy, where (i) *CFAPR* is used for recommending activity partners to users with past together preferences and (ii) *SIAPR* is used for the remaining users (recall that *SIAPR* performs best among the simple methods that do not rely on past together preference knowledge). We denote this hybrid method as *SICFAPR*. Figure 6 shows the result of *SICFAPR*, compared with using *SIAPR* to all users. Observe that *SICFAPR* exhibits constantly good performance on all tested cases.

5 Related Work

The most related work to ours includes recommendation approaches that also utilize social or user-profile information (Section 3.1) and work on collaborative filtering (Section 3.2). We also discuss related work on problems that are similar to activity-partner recommendation.



(a) Precision (Gowalla) (b) Recall (Gowalla) (c) Precision (Foursquare)

(d) Recall (Foursquare) (e) Precision (Brightkite) (f) Recall (Brightkite)

Fig. 6. Performance comparison of methods *SICFAPR* and *SIAPR* on all users (the ratio of #warm-start users to #cold-start users is about 2.0 (Gowalla), 1.5 (Foursqure) and 0.5 (Brightkite)).

Recommender Systems. Research on recommender systems in the previous years can be divided into two directions. The first is to improve existing models (e.g. collaborative filtering, content-based filtering, SVD based models) for recommendation. Another direction, which gains in popularity in the recent years, is to discover interesting applications of these models and extend base recommenders to domain-specific models and methods. Our work also falls in this direction. We study a new recommendation problem: recommend activity partners for the activity items suggested to a user.

Friend Recommendation. Recently, friend recommendation [11] became a popular research topic, assisting social networks to improve their service. 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) while our work explores (user, item)-user relationship (i.e., together preference from [user, activity item] to an activity partner). Friend recommendation estimates the likelihood that two non-friends will become friends in the future. Actually, the relatively bad performance of the *SCAPR* method, which employs the social closeness between users to recommend activity partners, verifies the intrinsic difference between friendships and activity partners.

Group Recommendation. Group recommendation [12] is to explore the preference of a group of users to items. Currently, many services (e.g., Movielens, Tencent QQ) allow users to create groups that consist of several users. Then, a typical objective of group recommendation is to aggregate the preferences from 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, 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).

6 Conclusion and Future work

In this paper, we have proposed and studied the problem of recommending activity partners to web-users for activity items suggested to them. Based on a questionnaire, we verify that real users have great interest in such a type of recommendation. We then show how to take advantage of different types of data and relationships, including attendance preference from users to activities, social context of users, and past together preference knowledge for activity-partner recommendation. Our experiments analyzed the strengths and weaknesses of the proposed activity partners recommendation models.

We have five directions in mind for future research. The first is to study how to combine the hypothesises introduced in this work into a hybrid component that considers all mentioned factors (social, attendance, and together preference) to rank the activity-partner candidates. The second is to investigate the effectiveness of activity-partner recommendation and the performance of *CFAPR* (and the other approaches tested in this paper) in additional application domains and with additional real-world data. Third, we plan to study how to combine together-preferences and attendance-preferences in order to adjust the ranking of activity items shown to the users. One idea would be to increase the ranking of activity items for which people can find more suitable activity partners. The fourth direction of future work is to integrate content information (e.g., the categories or geographical information of activity items) into our *CFAPR* framework, to see how far content information can improve activity-partner recommendation. Last, we also plan to study the problem of *Partner-Activity Recommendation*, where in friend recommendation we also include suggested activities for them to meet (e.g., the dating location and activity). This type of recommendation finds use in real-world applications, such as dating sites (e.g., www.jiayuan.com).

References

1. V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with gps history data. In *WWW*, pages 1029–1038, 2010.
2. L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.
3. X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.
4. M. J. Pazzani and D. Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. 2007.
5. J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. 2007.
6. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295, 2001.
7. Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
8. Q. He, J. Pei, D. Kifer, P. Mitra, and L. Giles. Context-aware citation recommendation. In *WWW*, pages 421–430, 2010.
9. G. Linden, B. Smith, and J. York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.
10. S. Wu, J. Sun, and J. Tang. Patent partner recommendation in enterprise social networks. In *WSDM*, pages 43–52, 2013.
11. J. Hannon, M. Bennett, and B. Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *RecSys*, pages 199–206, 2010.
12. J. Gorla, N. Lathia, S. Robertson, and J. Wang. Probabilistic group recommendation via information matching. In *WWW*, pages 495–504, 2013.