GroupMe: Supporting Group Formation with Mobile Sensing and Social Graph Mining

Bin Guo, Huilei He, Zhiwen Yu, Daqing Zhang, and Xingshe Zhou

Northwestern Polytechnical University, Xi'an, 710072, China
guobin.keio@gmail.com

Abstract. Nowadays, social activities in the real world (e.g., meetings, discussions, parties) are more and more popular and important to human life. As the number of contacts increases, the implicit social graph becomes increasingly complex, leading to a high cost on social activity organization and activity group formation. In order to promote the interaction among people and improve the efficiency of social activity organization, we propose a mobile social activity support system called GroupMe, which facilitates the activity group initiation based on mobile sensing and social graph mining. In GroupMe, user activities are automatically sensed and logged in the social activity logging (ACL) repository. By analyzing the historical ACL data through a series of group mining (group extraction, group abstraction) algorithms, we obtain implicit logical contact groups. We then use the sensed contexts and the computed user affinity to her logical groups to suggest highly relevant groups in social activity initiation. The experimental results verify the effectiveness of the proposed approach.

Keywords: Social graph mining, context-awareness, group formation and recommendation, mobile sensing, social activity organization.

1 Introduction

Forging social connections with others is the core of what makes us human. In modern life, people participate in various social activities each day. Depending on the distinct nature of a social activity, different crowds of people are involved. In this paper, we define the people participated in a social activity a *group* or a *clique*. For example, groups in a university can be project teams, dining partners, co-players, etc. The reason for the formation of distinct groups for different activities is that *people tend to be with a similar group of people to participate in certain activities*. Selecting members to form groups has thus become a significant step to organize social activities. This paper present our efforts for group formation in social activity organization, leveraging advanced mobile sensing and data mining techniques.

The design of a tool to facilitate social group formation is non-trivial. *The first challenge is how to accurately model and efficiently manage human groups*. For example, people usually participate in multiple groups with different roles, group size, and involved members. We use *social graph* to characterize the structure of a social activity participation network, which often consists of a set of overlapping and nested groups. For instance, *A* can be involved in both a sport team and a project team, the

K. Zheng, M. Li, and H. Jiang (Eds.): MOBIQUITOUS 2012, LNICST 120, pp. 200–211, 2013.

two relevant groups are thus overlapping); *A* has lunch with *B, C, D* one day and with *B, C* another day, the two activity groups are nested. This observation has prompted many social communication tools (e.g., Gmail Contact, friend manager in Facebook) that allow users to group their contacts. However, as investigated in previous studies [1, 2], group creation is time-consuming and tedious, and users of social communication tools rarely manually group their contacts. Furthermore, human relationships often evolve and groups change dynamically (e.g., having one member joined or removed). The creation of static social groups can thus quickly become stale. Supposing that implicit groups of a user can be extracted, *the second challenge becomes how to recommend highly relevant groups to the user when an activity is initiated*. It is affected by several factors, such as in-situ contexts (e.g., where the organizer locates, who is together with), adhesion of each group to the user, etc.

To lesson user effort on social activity organization and group management, it is beneficial to provide an intelligent application that can automatic category human groups and recommend relevant groups for a specific activity at hand (e.g., in terms of contexts). There have been recently several studies devoted to this (refer to Section 2 for details) [2, 3], which cluster and suggest contacts by virtue of analyzing historical interaction data among people. However, these systems are mainly focused on contact grouping and recommendation in online communities (e.g., emails, Facebook; typically used for formal or long-distance communication), they do not represent social activities in the real world, which are often formed in ad hoc, face-to-face manners. Comparing with online interactions, real-world interactions are more difficult to capture and record. For instance, there basically lacks a preexisting infrastructure (online interaction data can be maintained in mail servers or social web servers) for physical activity logging and mining. Furthermore, activity organization in the real world is often impacted by various social/personal contexts, which should be additionally considered when designing group recommendation algorithms.

To address the above issues, we have developed GroupMe, a group formation and recommendation tool that aims to facilitate social activity organization in the real world. Different from previous work that mainly works in online environments, we exploit sensor-enhanced mobile phones to capture human interactions and assist group formation in real world settings. Specifically, our contributions include:

- A social activity logging model, which depicts the major elements for real-world activities. We have also proposed the social graph, to characterize the social activity participation network at multiple granularities, e.g., raw/logical groups.
- A novel algorithm for automatic *group extraction* and *abstraction* from large-scale mobile sensing data, coupled with a user interface that can suggest contact groups, given the context of the user (e.g., user location, nearby people) and the estimated user affinity to the group.
- An evaluation of the quality and accuracy of our system. Results suggest that our algorithm models users' social activities sufficiently well, and can suggest contacts with high precision and recall.

The rest of this paper proceeds as follows: we first survey related work in Section 2; followed by the system architecture in Section 3; the activity logging and group models are described in Section 4; in Section 5, we present the core algorithms for group formation and recommendation; the prototype implementation and an evaluation of our system are described in Section 6; finally, we conclude the paper in Section 7.

2 Related Work

There are two closely related research areas of our work: *social interaction enhancement* and *group formation and management*.

2.1 Social Interaction Enhancement

Social interaction is important to human life and work. There have been numerous studies that aim to enhance human interaction and communication. One direction is to facilitate the management of the ever-increasing human contacts. For example, ContactMap provides an editable visualization of personal contacts, spatially organized and colored by group membership [4]. Our previous work, SCM (social contact manager) [5], is designed to automatically collect contact data and support efficient retrieval of human contacts based on associative cues. However, all these systems do not automatically group and suggest contacts, instead requiring manual layout and assignment of each contact.

Another direction is to enhance face-to-face human interactions. There have been studies that aim to improve social connectivity in physical communities by leveraging the information detected by mobile devices that contact. Social Serendipity is one of such studies, in which matching interests among nearby people who do not know one another are indicated as a cue for informal, face-to-face interactions [6]. The SOCKER application we developed is another example, which can build ad-hoc communities of like-minded people [7]. Though these systems can enhance the interaction among people with similar interests, they do not support mining and recommendation of groups to users based on their interaction history.

2.2 Group Formation and Management

Group formation and management is crucial for social activity organization and interpersonal communication. Researchers from Google have proposed a friend-suggestion algorithm, which can generate a recipient group when composing e-mails, given a small seed set of contacts [3]. MacLean et al. from Stanford University have developed a social group browser called SocialFlow [2], which can show social groups automatically mined from email data. These systems can extract social groups from online interactions and facilitate Web-based communication, but fail to address social activity initiation in real-world settings.

There have been quite few studies that devote to group formation in the physical world. For example, Flocks [8] is a system that supports dynamic group creation based on user profiles and physical proximity (e.g., forming a group with nearby badminton-lovers). MobilisGroups [9] is a location-based group creation service, which allows the user to initiate a social event on the map and recruit the ones using temporal and spatial filters (i.e., who is nearby at a given period of time). Though facilitating group formation in real-world settings, they mainly aim to group people who are already nearby and share certain commons, while not supporting the recommendation of contacts who are not yet gathered but should be, in terms of historical situations and in-situ contexts.

3 System Architecture

GroupMe aims to support social activity organization in the real world. There are two basic requirements: *(1) how to mine implicit groups from human interaction history in the physical world; (2) how to recommend highly relevant contact groups in terms of context*. We have designed a layered architecture to meet these requirements.

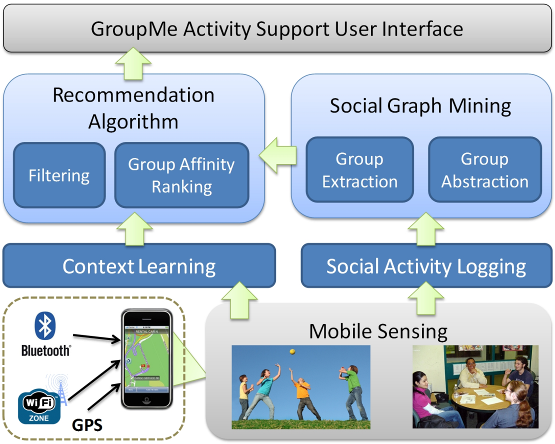


Fig. 1. GroupMe system architecture

The first layer is the *mobile sensing* layer, which consists of mobile phones enhanced by various sensors (e.g., Bluetooth, WiFi, GPS, accelerometers). Nowadays, mobile phones have become intimate "personal companions", which makes it possible to monitor human daily behaviors.

The second layer is the *data processing* layer, which involves two modules: context learning and social activity logging. The context learning module extracts *in-situ* user contexts from raw sensed data. The social activity logging module, nevertheless, transforms raw data to social activity logs and inserts into the SAL repository, according to the social activity logging model presented in Section 4.

In the third layer, the *group computing algorithm* layer, we have two components, *social group mining* and *recommendation*. The social group mining algorithm can extract and abstract logical groups from activity logging records. The recommendation algorithm can suggest highly relevant groups (mined from social group mining algorithm) to the user in terms of sensed contexts.

The fourth layer is *user interface*, which provides intelligent group formation and activity organization service with little manual effort.

4 Modeling Activity Organization and Group Formation

As shown in other studies, maintaining an interaction repository is the basis for group mining and suggestion. Distinct from online communication systems, where the interaction history has been kept in Web servers, social interaction in the real world should be captured and logged via a new way. In this section, we first present our definition of social activity; the social activity logging (SAL) model will then be presented; we finally describe the social graph and group model.

4.1 Social Activity Logging Model

Social activities can be held in physical, face-to-face or online/virtual manners. Here, we refer to traditional *meeting-based social activities (MSA)*, which can be defined as *a crowd of people that gather together at a preplanned time and place for a specific purpose*. Each activity MSA_i has its initiator, the initiation place, the activity venue, a group of activity members or participants. We use an example to illustrate it: *One day, Bob is in the laboratory and he wants to invite some friends to have dinner together in the Golf restaurant*. Here, Bob is the activity initiator; he initiates the activity in his laboratory; the activity venue is the Golf restaurant.

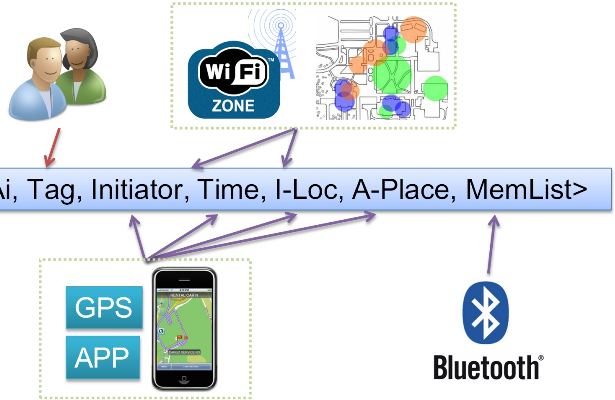


Fig. 2. The SAL model

According to the definition, we formulate the social activity logging (SAL) model, which describes how *MSAs* should be recorded in the data repository (we call it the SAL repository). It is illustrated in Fig. 2. The metadata are explained below.

- A_i: Activity index.
- *Tag:* Users may give one more tags for a activity, e.g., dinner, party, meeting.
- *Initiator*: The initiator of a social activity. We define the person who send out the activity invitation message as the initiator.
- *Time:* Activity initiation time.
- *I-Loc:* It refers to the location where the organizer initiates the social activity. For instance, the dinning activity may be initiated by Bob in his laboratory.
- *A-Place*: Place or venue of the social activity, e.g., the dinning activity may happen in the university restaurant.
- *MemList:* A list of members who participate the social activity. As group activities, we have: $Size[MemList] \geq 2$

Messaging and Logging. When an activity is initiazed, the initiator will send an invitation message (e.g., SMS) to a group of contacts. We category the message into two types: SA_{in} and SA_{out} . All the invitation messages will be kept in the initiator's SA_{out} box, the received activity requests will be kept in the SA_{in} box. For privacy, the *I-Loc* will be sent to empty to all the message receivers. People being invited (e.g., group members) can add new tags for incoming messages. All the logged messages (SA_{in} , SA_{out}) form the SAL repository.

4.2 Social Graph and Group Modeling

We use *social graph* to characterize the structure of a social activity participation network. Edges are formed by sending or receiving activity requests. We employ the egocentric network method used in [3], which considers a message sent from a user to a group of contacts as forming a single edge (a *hyperedge*). The edge is directed, represented as *in* and *out* edges (corresponding to SA_{in} and SA_{out}). We call each hyperedge an *explicit/raw group*. Figure 3 gives an example of *A*'s social graph, where three raw groups are involved (e.g., *G1* to *G3*). The directed edges are also illustrated in Fig. 3.

As presented in the introduction, the social graph of a person often consists of a set of overlapping (e.g., group *G1* and *G3*) and nested groups (e.g., group *G1* and *G2*). Here we also give a formulation of the two types of groups, as shown in Eq. (1), (2).

$$Overlap(G1, G2) = (G1 \cap G2 \neq \phi) \wedge (G1 \not\subset G2) \wedge (G2 \not\subset G1) \tag{1}$$

$$Nested(G1, G2) = (G1 \subset G2) \vee (G2 \subset G1) \tag{2}$$

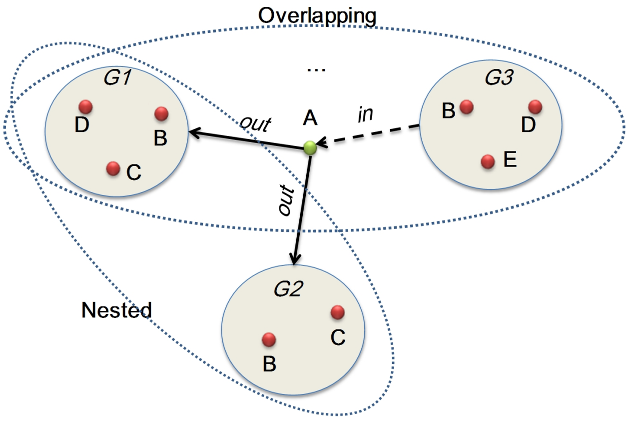


Fig. 3. Group formation and abstraction

People can participate in various social activities, and different social activities usually link different group instances. This will result in a large number of groups in the initial social graph. Different from the approach used in [3], which recommends groups based on the raw extracted groups. Our work introduces the *group abstraction* process, which can eliminate minor subsets of groups by merging highly nested or overlapping groups. For example, *A* has lunch with *B*, *C* and *D* each day, except for one day that *D* didn't come for some reason, this result in two different groups: *{A, B, C, D}* and *{A, B, C}*. In our approach, the two groups will be merged as the unique group: *{A, B, C, D}*. We call the groups after group abstraction *implicit/logical groups*.

5 Algorithms for Group Formation and Recommendation

Having described the approach for group modeling, in this section we first present the group abstraction process (i.e., *social graph mining*, distilling logical groups from raw extracted groups); the algorithm that can measure users' affinity to each logical group and support context-aware group recommendation will then be presented.

5.1 Group Abstraction

The group abstraction process is to merge highly nested/overlapping raw groups. We refer to the merging of nested groups as *group subsumption* and the merging of overlapping groups as *group integration*.

(A) Group Subsumption

Given two nested groups, *G1* and *G2* ($G1 \subset G2$). The two groups can be subsumed if they are highly nested. Here we refer to MacLean *et al.*'s *information leak* metric for group nesting evaluation [2]. The information leak value is determined by two factors: *similarity of the two groups*, and *the ratio of the number of social activities held by each group* (e.g., the number of records in SAL). We thus define a new parameter *subrate*, to measure if two groups can be subsumed. This is formulated in Eq. (3).

$$subtrate(G1, G2) = \frac{|G2| - |G1|}{|G2|} \times \frac{num(G1)}{num(G2)}, \quad when\ G1 \subset G2 \tag{3}$$

Where: $|Gi|$ refers to the number of members of group *Gi*, and $\frac{|G2| - |G1|}{|G2|}$ characterizes the similarity of two groups; *num(Gi)* refers to the number of social activities held by *Gi*. Suppose *G1= {A, B}*, *G2={A,B,C}*, and there are 5 and 100 records relevant to *G1* and *G2* in SAL, we have $subtrate(G1, G2) = \frac{3-2}{3} \times \frac{5}{100} = 1/60$. If the value is below a predefined threshold (*subThreshold*), the two groups can be subsumed.

(B) Group Integration

The two overlapping groups can be integrated if they are very similar. To measure the similarity between two groups, we use the *Jaccard* metric which is often used for similarity measurement [10, 11]. A new parameter called *intrate* is defined, formulated in Eq. (4). The two groups can be integrated if their similarity exceeds a threshold (*intThreshold*).

$$intrate(G1, G2) = \frac{|G1 \cap G2|}{|G1 \cup G2|}, \quad when\ overlap(G1, G2) \tag{4}$$

5.2 Group Recommendation

Having identified implicit groups, the next requirement is to recommend highly relevant groups to users in real-world settings. The recommendation is based on two major factors: *the context of the user*, and *the affinity between the user and his groups*.

The algorithm is thus designed by two major parts: *context-aware group filtering* and *group affinity ranking*.

(A) Context-Aware Group Filtering

One basic principle for group recommendation is to suggest relevant groups in terms of user needs with little human intervention. Various contexts that are obtained when users initialize activities are leveraged to filter irrelevant logical groups.

- *Time*: we divide the initiation time into four logical period of times, namely *morning* (6:00-11:00), *noon* (11:00-13:00), *afternoon* (13:00-18:00), *night* (18:00-6:00).
- *Location*: the location where the user initiates an activity (e.g., *I-Loc*). It can be obtained through *in-phone* GPS positioning or WiFi positioning.
- *WithWhom*: nearby friends are often co-initiators or members of an activity. We use *WithWhom (i)* to represent that a number of *i* contacts are together with the initiator. This context can be obtained through the Bluetooth ID of user mobile phones, and a user will keep the Bluetooth ID of her friends in her contact book.
- *Tag*: a tag given by the user often shows the type of an activity being organized.

The rule for group filtering is performed in this way: for each context *Ci* obtained when organizing a new activity, if a logical group *Gj* does not have any historical record (as depicted in Section 4.1, each logical group corresponds to a set of historical records in SAL) that matches *Ci*, *Gj* is considered irrelevant and thus will be filtered.

(B) Group Affinity Ranking

Group affinity ranking is to calculate the tie strength between a user and her logical groups. There have been numerous studies on tie-strength evaluation in social networks [12, 13, 3]. Here, we employ the method used in [3], which is originally used for contact tie-strength measurement in email networks. The tie strength between two entities is computed based on their interaction history. Besides interaction frequency, two other factors are considered:

- *Recency*. Human relationship is evolvable and dynamic over time.
- *User role.* The social activities that the user initiates (i.e., as the initiator) are considered more important than those he or she is merely a participant.

We define the affinity rank between user *Ui* and logical group *Gj* as *affrank(Ui, Gj)*, which can be computed by Eq. (5) :

$$affRank(Ui,Gj) = \omega_{out} \sum_{Ai \in SA_{out} \wedge G(Ai)=Gj} (\frac{1}{2})^{d_{now}-d(Ai)} + \omega_{in} \sum_{Ai \in SA_{in} \wedge G(Ai)=Gj} (\frac{1}{2})^{d_{now}-d(Ai)} \quad (5)$$

Where: ω_{out} and ω_{in} weight the user roles in social activities, the former one is bigger to represent the importance of initiator roles. We use empirically 1.5 and 1.0 in the current implementation.

SA_{out} and SA_{in} follow the definition in Section 4.1, indicating initiated activity records and being invited activity records; $Ai \in SA_{out} \wedge G(Ai) = Gj$ means that the activity record *Ai* is from SA_{out} and the corresponding group of *Ai* is *Gj*.

d_{now} and *d (Ai)* refer to the current date and the initiation time of activity *Ai*.

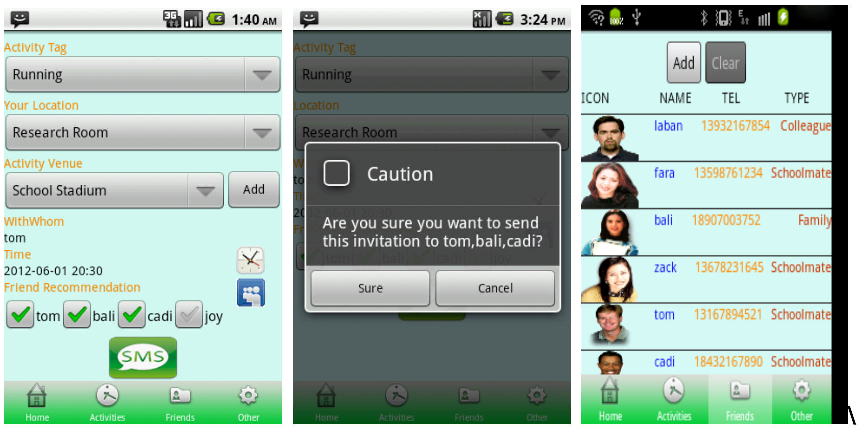
Given *Ui*, the implicit group with the highest rank will finally be recommended.

6 Implementation and Evaluation

In this section, we first present a prototype implementation of GroupMe. The implementation of GroupMe is based on several key components, such as context-aware recommendation and group abstraction. We will then evaluate the affects of these components to the performance of GroupMe. An initial user study was also conducted to validate the usability of our system.

6.1 Prototype Implementation

We have implemented GroupMe on the Android platform. The SQLite was used as the database for activity data storage. The interface for activity organization is shown in Fig. 4 (a). The *location*, *initiation time*, and *WithWhom* context can be sensed automatically; tags and activity venue can be added by users for activities. Recommended friends are also listed. When pressing the ‘SMS’ button, an SMS message will be sent out to the selected contacts (Fig. 4b). The contact management interface is shown in Fig. 4 (c), where user profile can be managed.



(a) (b) (c)

Fig. 4. User interfaces: (a) activity organization, (b) invitation sending, and (c) contact management

6.2 Data Collection

As an intelligent system based on social interaction history mining, data collection becomes the basis for system performance evaluation. In the current stage, a combination of two methods can be used for data collection: *mobile phone logging* and *online blogging*. The prior method automatically logs user activities when they initialize them using the GroupMe software (installed in sensor-enhanced mobile phones). The latter one asks users to manually record their daily social activities in an online blogging webpage.

Since it is not easy to equip a smart phone to each data contributor, the online blogging method was used chiefly in the data collection process. Twenty more students from our lab were recruited to contribute data, the data collection activity lasted for about one month during April-May, 2012. Almost four hundred activity records were collected, among which the five most popular activities recorded are *lunch (38.3%)*, *lesson (16.8%)*, *discussion (13.4%)*, *sports (6.15%)*, *meeting (5.6%)*. The two most popular initiation places are *lab* and *student dormitory*. We plan to collect more data using mobile phone logging in the next stage.

6.3 Performance Evaluation

(A) Evaluation Metric and Parameter Setting

To validate the effectiveness of the recommendation algorithm, we employ two generic criteria — *Precision* and *Recall*. *Precision* is the ratio of the correct number of recommendations (*RightRecNum*) and the total number of recommendations (*TotalRecNum*). Recall is the ratio of *RightRecNum* and the number of people who are actually invited (*ActMemberNum*), i.e., the *ground truth*. The two criteria are formulated by Eq. (6) and (7).

$$precision = \frac{RightRecNum}{TotalRecNum} \tag{6}$$

$$recall = \frac{RightRecNum}{ActMemberNum} \tag{7}$$

For instance, suppose *A*, *B* and *C* participate an activity, the recommendation result of GroupMe is *A*, *B*, *D* and *E*. Here, the *RightRecNum*, *TotalRecNum*, and *ActMemberNum* are 2, 4, and 3, respectively. The *Precision* of the recommendation is thus 50% and the *Recall* is 67%. In the experiments, we choose 300 SAL records as the training set, and 50 as the test set. The *MemList* in the test records are used as the ground truth, and the training set is used for computing recommendations. In the experiments, the *subThreshold* and *intThreshold* are set empirically to 0.2 and 0.3.

Table 1. The effect of contexts

Context Groups	Precision	Recall
(Initiation) Time + I-Loc	58.2%	74.6%
Time + I-Loc + Tag	71.2%	80.7%
Time + I-Loc + WithWhom (1)	68.13%	94.7%
Time + I-Loc + WithWhom (2)	81.01%	98.66%

(B) The Effect of Contexts

One of the major differences between GroupMe and other group tools is that our work is to provide group suggestion in pervasive, real-world settings. Many contexts obtained through mobile sensing are leveraged to filter irrelevant groups and improve recommendation performance. To evaluate the effects of different contexts to group suggestion, we have chosen four different groups of contexts, with *Time* and *I-Loc* as the basic group, and *Tag*, *WithWhom(1), WithWhom(2)* as additional elements in the other three context groups. The experiment results are listed in Table 1, which shows that more contexts can enhance recommendation performance, and the *WithWhom* context performs better than the *Tag* context.

(C) The Effect of Group Abstraction

Group abstraction is another contribution to group formation and suggestion, which can eliminate noisy groups and merge relevant raw groups to logical units. We have conducted experiments to validate its effect to GroupMe, by comparing the *Precision* and *Recall* of the recommendation with and without group abstraction. Three contexts are used: *time*, *I-Loc*, and *WithWhom (0, 1, 2)*. The experimental results are shown in Fig. 5, which indicates that group abstraction can better draw the social graph of a user and provide more effective support for activity organization.

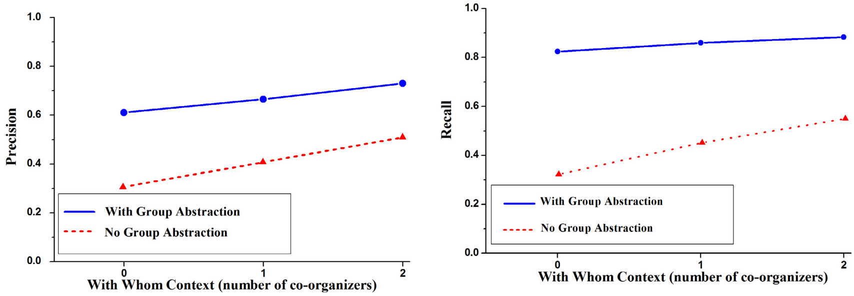


Fig. 5. The effect of group abstraction

(D) User Study

To understand the usability of our system, we have made a user study to compare the efficiency of activity organization in traditional flat contact lists and in GroupMe. Ten subjects who have contributed in the data collection process were recruited for this study. For each of them, we specified an "activity tag" (according to her activity records in the SAL repository), and asked the subject to invite the people who often participate this activity together with her. The flat contact list contains a list of 100 people. For GroupMe, three contexts were used for contact recommendation: *Time*, *I-Loc*, and *WithWhom(1)*. Once the subject specified the three contexts, the recommendations were given. The average time used for the two methods is 12s (for flat contact list) and 3s (for GroupMe), which illustrates that GroupMe can save much time on group formation.

7 Conclusion

We have presented our early efforts for social activity organization in real world settings. The activity logging and social graph model is proposed to characterize meeting-based social activities and complex, heterogeneous group structure in activity participation. A series of group computing algorithms are presented to extract logical groups from raw groups. To suggest highly relevant groups, the context and adhesiveness-aware algorithm are proposed. Experiments over the one-month activity logs collected from 20 more subjects show that, by using various contexts and the group abstraction process, the performance of group formation and suggestion can be improved. The user study indicates that our system greatly decreases the time cost on group formation than traditional ways. Social activities and human behaviors are difficult to model due to its complex nature. For instance, people sometimes want to have activities with close friends, and sometimes they intend to make new friends. As for future work, we intend to extend the system to involve more group formation methods (e.g., not only mining existing groups, but suggesting new contacts to join). We will leverage the opportunistic contact nature [14] and social network structure (e.g., triadic closure [15]) theories to achieve this.

Acknowledgement. This work was partially supported by the National Basic Research Program of China (No.2012CB316400), the National Natural Science Foundation of China (No. 61222209, 61103063), the Program for New Century Excellent Talents in University (No. NCET-12-0466), the Natural Science Basic Research Plan in Shaanxi Province of China (No. 2012JQ8028), and the Basic Research Foundation of NPU (No. JC20110267).

References

1. Kuhn, M., Wirz, M.: Cluestr: Mobile social networking for enhanced group communication. In: Proc. of the International Conference on Supporting Group Work, GROUP (2009)
2. MacLean, D., Hangal, S., Teh, S.K., Lam, M.S., Heer, J.: Groups without tears: mining social topologies from email. In: Proc. of IUI 2011, pp. 83–92 (2011)
3. Roth, M., et al.: Suggesting friends using the implicit social graph. In: Proc. of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2010)
4. Whittaker, S., et al.: ContactMap: Organizing communication in a social desktop. ACM Transactions on Computer-Human Interaction (TOCHI) 11(4), 445–471 (2004)
5. Guo, B., Zhang, D., Yang, D.: “Read” More from Business Cards: Toward a Smart Social Contact Management System. In: Proc. of WI 2011, pp. 384–387 (2011)
6. Eagle, N., Pentland, A.: Social Serendipity: Mobilizing Social Software. IEEE Pervasive Computing 4(2), 28–34 (2005)
7. Zhang, D., Wang, Z., Guo, B., Raychoudhury, V., Zhou, X.: A Dynamic Community Creation Mechanism in Opportunistic Mobile Social Networks. In: Proc. of the Third IEEE International Conference on Social Computing (SocialCom 2011), pp. 509–514 (2011)
8. Boix, E.G., et al.: Flocks: enabling dynamic group interactions in mobile social networking applications. In: Proc. of SAC 2011, pp. 425–432 (2011)
9. Lubke, R., Schuster, D., Schill, A.: Mobilisgroups: Location-based group formation in mobile social networks. In: Proc. of PerCom Workshops, pp. 502–507 (2011)
10. Tan, P., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison Wesley (2005)
11. Liben, N.D., Kleinberg, J.: The link prediction problem for social networks. In: Proc. of CIKM 2003, pp. 556–559 (2003)
12. Gilbert, E., Karahalios, K.: Predicting tie strength with social media. In: Proc. of CHI 2009, pp. 211–220 (2009)
13. Xiang, R., Neville, J., Rogati, M.: Modeling relationship strength in online social networks. In: Proc. of WWW 2010, pp. 981–990 (2010)
14. Guo, B., et al.: Enhancing Spontaneous Interaction in Opportunistic Mobile Social Networks. Communications in Mobile Computing (ComC) 1(6) (2012)
15. Kleinberg, J., Easley, D.: Networks, Crowds, and Markets. Cambridge University Press (2010)