

Social Recommendations for Events

Toon De Pessemer, Jeroen Minnaert, Kris Vanhecke, Simon Doods, Luc Martens

iMinds - Ghent University
Dept. of Information Technology
G. Crommenlaan 8 box 201
B-9050 Ghent, Belgium

{Toon.DePessemer, Jeroen.Minnaert, Kris.Vanhecke, Simon.Doods, Luc1.Martens}@UGent.be

ABSTRACT

Due to an abundance of available events, selecting the most interesting events and deciding who to invite to attend these events becomes increasingly difficult. This paper describes the Outlife recommender, which assists in finding the ideal event by providing recommendations based on the user's personal preferences. For each event recommendation, the Outlife recommender suggests a group of friends to invite to the event. Conversely, the user can select a group of friends and receive group recommendations based on the preferences of all group members. Compared to traditional group recommenders, which assume the group composition is available as input, the Outlife recommender automatically composes the user's groups of best friends based on interaction behavior. A user evaluation showed that the composition of groups of friends is accurate and that the offered recommendations match the user's preferences.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Information Filtering; H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces

General Terms

Algorithms, Experimentation

Keywords

group recommendation, event, social recommendation, clustering users

1. INTRODUCTION

Research in the domain of recommender systems mainly focuses on solutions for online shops or video-on-demand services. Still, the application area of (cultural) events is less exploited and the specific characteristics of events (such as

the limited availability in time and space) do not always allow to use traditional, general-purpose solutions. The time-specific restrictions of events can be handled by using a hybrid recommendation technique [1], combining content-based and collaborating filtering approaches. Moreover, an online user-centric evaluation revealed that for an event recommender, a hybrid approach provides the best results in terms of accuracy, novelty, usefulness, and satisfaction [2]. Another study showed that for events that have not taken place and for which no direct feedback exists, recommendations can be generated based on individuals' preferences for past events, combined collaboratively with other peoples' likes and dislikes [6].

To compensate for the lack of feedback for future events, the user's social network can be used as an additional knowledge source. In the context of academic events (conferences, workshops, international symposiums, etc.), social network analysis and recommender systems were combined to assist the user in finding papers, books and experts [4].

Attending a cultural event is typically a social and joint activity: people often go to museums, movie theaters, workshops, concerts, and fairs together with friends or family. As a result, group recommendations considering the preferences of all people that plan to attend the event, are more suitable than the traditional single-user recommendations. Most group recommender systems assume that the groups are composed in advance and are available as input for the recommender [3, 5]. However, the group composition can be dynamic and change according to the availability of the group members, their interests in the event, their current location, etc.

So in case of dynamic groups, a new challenge for group recommenders is the automatic composition of the groups. This paper describes how groups of users can automatically be composed and offered to the user along with the recommended items as output of the recommender system. More specifically, this paper presents the Outlife recommender system, which focuses on two scenarios:

1. Generating event recommendations based on the individual preferences of the user, together with a selection of the user's Facebook friends who would be happy to join the user at the recommended event.
2. Composing groups consisting of the user's best Facebook friends who belong together; and recommending events for each of these groups according to the preferences of all group members.

RecSys'13, October 12–16, 2013, Hong Kong, China.

Paper presented at the 2013 RSWeb@RecSys workshop in conjunction with the 7th ACM conference on Recommender Systems. Copyright ©2013 for the individual papers by the papers' authors. Copying permitted for private and academic purposes. This volume is published and copyrighted by its editors.

2. RECOMMENDING GROUPS

2.1 Clustering Friends

In order to suggest a group of friends to attend an event together with the user of the system, the user’s Facebook friends are clustered based on their social network. Because the number of groups, or in other words the number of clusters, is not known in advance, a hierarchical clustering technique is used. More specifically, groups of friends are composed by using agglomerative hierarchical clustering, a bottom-up approach that starts with each friend as an individual cluster and iteratively merges the pair of clusters that are the closest to each other. Given the limited number of friends of a typical Facebook user (on average 190 Facebook friends [9]), the bottom-up approach is more efficient than the alternative top-down approach.

Clustering techniques are based on a distance metric, a measure of the distance between pairs of observations. In agglomerative hierarchical clustering, this distance metric is used to decide which two clusters have to be merged into a new cluster. To cluster the user’s Facebook friends, a distance metric based on mutual friends is employed. (Facebook allows to query the mutual friends of a user and a friend of that user.)

$$d_{a,b} = |(Members_a \cup Mutual_{you,a}) \cap (Members_b \cup Mutual_{you,b})| \quad (1)$$

$$Mutual_{c,d} = Friends_c \cap Friends_d \quad (2)$$

The reasoning of this distance metric is as follows. If a friend of you has a lot of friends in common with another friend of you, then these two friends probably belong to the same group of friends. Indeed, people who spend a lot of time together probably meet the same people. In contrast, if two friends have no friends in common, then these two friends will probably belong to a different group.

The merging of clusters that are the closest to each other is stopped as soon as the distance metric reaches a certain threshold. Experiments indicated that the optimal threshold value is $d_{a,b} = 0$, or in other words, the merging of clusters is stopped if the clusters have no member or mutual friend in common.

2.2 Selecting Best Friends

The groups of friends that are composed by the clustering algorithm can be quite large (up to 100 members). Since people generally attend an event in smaller groups (typically with less than 10 friends), only the user’s best friends are selected for joining the event. Therefore, the members of a cluster are ranked according to their friend score, and the user receives a recommendation to attend the event with ten friends (of a single cluster) who achieved the highest ranking.

The friend score is calculated based on indicators of social interactivity on Facebook, more specifically, *likes* and *tags* on *photos*, *posts*, and *check-ins*. A friend can like a photo, post, or check-in thereby showing appreciation for the Facebook item. A like may indicate some interest of the user for the photo, post, or check-in of the friend but provides no evidence for a connection between the user and that friend in the real life.

A Facebook tag is a reference to a particular person in a Facebook item. It establishes a direct link between the

Table 1: Weights of the likes and tags on photos, posts, and check-ins

	Like	Tag
Check-in	1	50
Post	1	25
Photo	1	15

tagged friend and the item (and as a result also the user who published the item). Therefore, tags can be considered as more profound interaction between users than likes. Tags on different types of items also have a different connotation regarding the interaction of two users. Tagging a friend on a check-in signifies the joint presence of the user and the friend at the location of the check-in, thereby indicating that these two persons hang out with each other. As a result, a tag of a friend on a check-in is a strong indication that this friend might join the user for future events too. A tag in a post, i.e., an explicit reference to a friend in the user’s status update, expresses a link between the two people but does not mean that these two persons attend events together. A tag of a friend in a photo can indicate that the user spends time with this friend. Unfortunately, tags in photos introduce some uncertainty, since two persons who have no link to each other are sometimes tagged in the same photo on Facebook. Moreover, in case of a photo in which a lot of people are tagged, the importance of the link between two persons in the picture decreases. Based on these assumptions and the relative frequency of the various interactions, the weights of the interactions, which reflect their importance, are determined and listed in Table 1.

Based on these weights, the Friend Contribution FC is determined for each friend a and each Facebook item i .

$$FC_{a,i,j} = \log\left(\frac{1}{time_i + 1}\right) \cdot weight_{i,j} \begin{cases} i \in \text{check-ins} \cup \text{posts} \\ j \in \text{likes} \cup \text{tags} \end{cases} \quad (3)$$

Here, $time_i$ stands for the age (expressed in weeks) of the Facebook item. Since friendships can dilute over time, interactions on older Facebook items have a lower contribution to the estimated friendship of two people. The logarithm reflects that friendships dilute fast at the first weeks, but at a lower rate after a long period of time.

In a photo of a group of people, the link between a user and a friend is less convincing, especially if a lot of people are tagged in the photo. Therefore, the friend contribution is divided by the number of people that are tagged in the photo.

$$FC_{a,i,j} = \frac{\log\left(\frac{1}{time_i + 1}\right) \cdot weight_{i,j}}{\text{number of people in the photo}} \begin{cases} i \in \text{photos} \\ j \in \text{likes} \cup \text{tags} \end{cases} \quad (4)$$

Finally, the Friend Score $FS_{u,a}$ of the user’s friend a is calculated as the sum of the friend contributions for all Facebook items of user u . This friend score is used to select the best friends of a user from each cluster.

$$FS_{u,a} = \sum_{i \in \text{items}_u, j \in \text{likes} \cup \text{tags}} FC_{a,i,j} \quad (5)$$

3. RECOMMENDING EVENTS

For each user or each group of best friends as calculated in Section 2, recommendations can be calculated according

to the preferences of the user and the friends. These recommendations are calculated in four phases.

- 1 First, all the appropriate events are selected, meaning only events within a specific geographic region, time period, and price range.
- 2 Then, for each user and each individual member of the group, event recommendations are calculated. For registered users of the Outlife recommender, of whom the preferences can be derived from their ratings, the Duine framework [8] is used. This recommendation framework consists of a set of algorithms (content-based as well as collaborative filtering approaches) and a decision tree to select the most appropriate algorithm for each situation. For unregistered users, e.g., the user’s Facebook friends who never used the Outlife recommender, no ratings are available. For these users, recommendations are calculated by a content-based filter based on their likes for events on Facebook and the categories of these events.
- 3a In case the recommendations are intended as **personal** suggestions for the user, the preferences of the user’s friends are compared with the recommended events. For each recommended event, the most appropriate groups of friends are selected based on their calculated rating prediction. These groups of friends, who might also like the event and might be willing to join the user, can be invited by the user. Suggestions for inviting these friends are added as supplementary information to the event recommendations.
- 3b In case the recommendations are intended for a predefined **group** of friends, the recommendations of each individual group member are aggregated into a list of group recommendations, which (hopefully) satisfies all group members. For each event, the algorithm’s rating predictions of each group member are aggregated using the *average without misery* strategy [5]. This strategy calculates the rating prediction for the group as the average of the individual rating predictions, but without events that score below a certain threshold for individuals.
- 4 The goal of the last phase is to introduce serendipity into the final recommendation list. Therefore, the recommendations are partitioned into different classes according to their rating prediction. From the first class, which contains the events with the highest rating prediction, $N - n$ events are selected for the final (group) recommendations. The remaining n recommendations are uniformly selected from the remaining classes, which might contain less accurate but more diverse and serendipitous events. Finally, the union of the selected recommendations, i.e., the resulting N events, are presented to the users.

4. THE RECOMMENDER SYSTEM

The Outlife recommender is implemented as a client-server system. The server is responsible for clustering the user’s friends, selecting the user’s best friends, calculating the recommendations, and storing the data about the user and feedback for events. Moreover, the server retrieves data about the available events from online information providers, such as the Flemish CultuurNet¹. This way, more than

¹<http://www.cultuurnet.be/>

100,000 events a year are available to explore and attend with friends.

The client is an iPhone application, which is illustrated in Figure 1 by two screen shots. The main screen contains four tabs named “My Events”, “Group Events”, “Bookmarks”, and “My Profile”. As shown in Figure 1(a), the first tab presents the individual recommendations according to the user’s personal preferences. For each recommended event, the user receives suggestions for groups of friends who probably like to join the user at the event. The event together with a picture, title, location, category, and date are displayed at the top. The recommended events can be browsed by swiping this area of the touch screen. At the bottom, a group of friends is represented by means of the profile pictures of these friends. The colored ribbon on each profile picture, which ranges from red to green, indicates how much the recommended event matches the preferences of the friend. The suggested groups can be browsed by swiping this area of the screen. Below the event information, two pairs of like/dislike buttons are available: one pair to evaluate the recommended event, and one pair to evaluate the match between the recommended event and the group of friends.

The second tab, “Group Events”, shows for each group of friends a set of events that probably match the group’s preferences. These recommended events are not only based on the preferences of the user of the application but also on the preferences of all friends of the user that are a member of the group.

The third tab, “Bookmarks”, allows users to consult the events that he/she has bookmarked in the application. Via the fourth tab, “My Profile”, users can consult and update their explicitly specified preferences (initial profile).

In addition, users can also specify some extra settings for the recommender system, and thereby fine-tune their personal preferences. Figure 1(b) shows that the user can set the maximum distance to an event (Radius), the number of recommended events (Predictions), the number of recommended groups that match an event (Groups), and how far in the future a recommended event can be (Days).

5. EVALUATION

5.1 Group Selection

To evaluate the clustering of friends into groups and the selection of the user’s best friends, an online user test was conducted in which 45 users, recruited from Facebook, had to evaluate the results of the algorithm using an online questionnaire. On average, the test users had 316 Facebook friends, of which the algorithm of Section 2.2 selected the 10 best friends based on their interaction behavior. 64% of the test users agreed with the calculated list of their best friends, meaning that the 10 selected friends match their best friends. More importantly, 71% of the users had actually attended one or more events together with 7 or more people of the list and 62% of the users plan to invite 7 or more people from the list. So the user’s best friends as calculated by the algorithm are mostly people who have already joined the user at an event as well as people who will probably be invited by the user to attend a future event.

In addition, user were asked to evaluate their groups of friends, which are calculated using the algorithm of Section 2.1. The 15 best friends of each of the 10 biggest clusters were presented to the user. For each friend, test users were



Figure 1: Screenshots of the mobile application: (a) an event recommendation with a group of friends to invite and (b) the possibility to filter the events on different aspects

asked to specify if the friend actually belongs to the group to which he/she was assigned by the algorithm. On average, 84% of the friends were classified in the correct group according to the test users, indicating that the algorithm is able to identify the user’s social circles. Furthermore, users were asked for each group how many of these group members they would invite to an event. On average, 54% of the group members would be invited by the users to an event of their choice. For some groups, all users would be invited, while for other groups nobody would be invited. This corresponds to a realistic scenario, since in general not all groups of friends are part of the user’s social life.

5.2 Event Recommender

An in-depth evaluation of the Outlife recommender was conducted by means of a user test in which 6 people, recruited by using a convenience panel-sampling method, were asked to try the application on their iPhone for one week. After using the application, they were asked to fill in an online questionnaire with 21 questions that were based on the user-centric evaluation framework of Pu et al. [7]. The users confirmed that the recommendations match their personal preferences and that they notice an accuracy improvement for the recommendations as well as for the groups after using the feedback buttons. Moreover, the test users experienced the recommendations as surprising and novel, indicating that the fourth phase of the algorithm adds the necessary serendipity to the recommendations while still being accurate.

Test users generally had a good feeling about the recommender system. They clearly appreciated the recommended events and groups as well as the possibility to provide feedback on the events and groups. On the downside, test users complained about the fact that events sometimes did not fit the target audience, e.g., in terms of age. Finally, test users preferred to see the recommender system integrated in an existing service, such as Foursquare ², thereby fully

²<http://foursquare.com>

exploiting the added value of the recommender system.

6. CONCLUSIONS

The Outlife recommender offers users personalized suggestions for events, as well as suggestions for inviting a group of friends to attend the recommended event together. Conversely, the user’s groups of friends can automatically be composed, and the most suitable events are then recommended based on the preferences of all group members. Groups of close friends of the user are composed by clustering the users based on their mutual friends and selecting the user’s best friends based on interaction behavior. Complementing the top-n recommendations with events that obtained a lower rating prediction enhances the serendipity and the user satisfaction. Future research can comprise the labeling of the detected groups of friends, incorporating user trust in the group recommendations, tackling the cold-start problem for new users, and a detailed user study over a longer period of time.

7. REFERENCES

- [1] C. Cornelis, X. Guo, J. Lu, and G. Zhang. A fuzzy relational approach to event recommendation. In *Proceedings of the Indian International Conference on Artificial Intelligence*, pages 2231–2242, 2005.
- [2] S. Dooms, T. De Pessemier, and L. Martens. A user-centric evaluation of recommender algorithms for an event recommendation system. In *Proceedings of the workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces at ACM Conference on Recommender Systems (RECSYS)*, pages 67–73, 2011.
- [3] A. Jameson and B. Smyth. Recommendation to groups. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 596–627. Springer Berlin Heidelberg, 2007.
- [4] R. Klamma, P. Cuong, and Y. Cao. You never walk alone: Recommending academic events based on social network analysis. In J. Zhou, editor, *Complex Sciences*, volume 4 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 657–670. Springer Berlin Heidelberg, 2009.
- [5] J. Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Modeling and User-Adapted Interaction*, 14:37–85, 2004.
- [6] E. Minkov, B. Charrow, J. Ledlie, S. Teller, and T. Jaakkola. Collaborative future event recommendation. In *Proceedings of the 19th ACM international conference on Information and knowledge management, CIKM '10*, pages 819–828, New York, NY, USA, 2010. ACM.
- [7] P. Pu, L. Chen, and R. Hu. A user-centric evaluation framework for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems, RecSys '11*, pages 157–164, New York, NY, USA, 2011. ACM.
- [8] Telematica Instituut / Novay. Duine Framework, 2009. Available at <http://duineframework.org/>.
- [9] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. The anatomy of the facebook social graph. *arXiv preprint arXiv:1111.4503*, 2011.