A Consensus-Focused Group Recommender System

Stratis Ioannidis
Technicolor,Palo Alto
stratis.ioannidis@technicolor.com

S. Muthukrishnan
Rutgers New Brunswick
muthu@cs.rutgers.edu

Jinyun Yan*
Rutgers New Brunswick
jinyuny@cs.rutgers.edu

ABSTRACT

In many cases, recommendations are consumed by groups of users rather than individuals. In this paper, we present a system which recommends social events to groups. The system helps groups to organize a joint activity and collectively select which activity to perform among several possible options. We also facilitate the consensus making, following the principle of group consensus decision making. Our system allows users to asynchronously vote, add and comment on alternatives. We observe social influence within groups through post-recommendation feedback during the group decision making process. We propose a *decision cascading model* and estimate such social influence, which can be used to improve the performance of group recommendation. We conduct experiments to measure the prediction performance of our model. The result shows that the model achieves better results than that of independent decision making model. The demo is accessible at `http://tinyurl.com/groupprecsys`.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, System, Experimentation

Keywords

Group Recommendation, Group Consensus Decision Making, Social Influence

1. INTRODUCTION

Existing recommender systems, for example, Netflix and Amazon, mainly target individuals. However, people engage in *social* events as a group both offline and online, e.g., movie watching, online gaming, dining out, book reading, and so on. The inherent need for social events and the large content space have inspired recent interests in group recommendation. To date, a variety of early stage systems have been developed for different applications, such as group web page recommendation [7], vacations or tours to groups of tourists [2], music tracks and playlists to large groups of listeners [6], and movies and TV programs to friends and family [4, 13].

---

*Authors by alphabetical order.

Prior work on group recommendation mainly focuses on the modeling of group preference. The intuition is that if we can infer group preference, the problem is reduced to individual recommendation. PolyLens [4] uses a "pseudo-user" to represent the group's preference. The "pseudo-user" is either created manually by group members or automatically merging the rating history of the group members. Besides averaging individual ratings, the paper [8] analyzed many other aggregation strategies, such as least misery, fairness, plurality voting and so on. Amer-Yahia *et al.* [1] defined a consensus function of the group which combines the aggregation and variance of individual ratings. Another approach is first obtaining the results of individual recommendations, then merging them to a group list which minimizes its distortion to individual lists. Baltrunas *et al.* [3] applied several rank aggregation methods to merge recommendation results of individuals.

Unfortunately, the post-recommendation behavior is usually neglected. In reality, when a group of people participates a joint event, individuals collectively make the choice over a set of alternatives of which activity to perform. Members and the social group itself possess social influence which contributes to the outcome, which is often different from those made by individuals. The group decision making process includes the interaction among members. Members get to know what other members like and dislike, and may decide either to conform to other members' preference or persuade them. As opposed to individual recommendation, the recommender system can easily figure out a person's reaction to the recommendation by a single click or a score. We argue that group recommender system should not ignore this process to learn the group and the members' feedback to group recommendation. Such feedback can be used to measure the recommendation and improve the accuracy of the profiling of the group.

Existing systems either only display group recommendation results to each individual as in [4, 11], or assume members can discuss alternatives offline [9]. In our system, we display group recommendation results to each individual, and also allow members to discuss and decide on alternatives either simultaneously or asynchronously. We adopt voting as the consensus decision making mechanism. The interface design of the system follows the principle of group consensus decision making [5], and the framework of the group decision is in Figure 1. The recommendation algorithm will contribute on proposal phrase, that occurs both before and during the
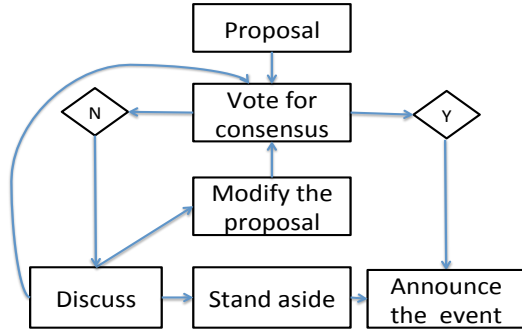
group decision making.

Figure 1: Group Decision Making

Members can vote in arbitrary order. When a member enter the decision process, other members may have already decided their preference. The previous decisions may influence the member. Social influence occurs when one's decision is affected by others. We observe social influence through the individual feedback during the discussion process. An individual votes positively for an item according to his/her own preference on the item and other members' decisions. In this paper, we propose a decision cascading model to estimate the social influence among members. We conducted experiments which show that using social influence can better predict the individual feedback in the voting process. We then propose the influence model to rank items for the group.

The major contributions of our work are as follows.

- We build a group recommender system taking dining out events as the application. The system can generate recommendations suited for groups.
- We design an interface that supports asynchronous voting, adding and commenting on alternatives.
- We infer the social influence among members through their feedback in the consensus discussion process.

2. THE DEMO SYSTEM

We present our group recommender system. The demo is accessible at `http://74.95.195.230:8889/`. Figure 2 shows the screenshots of important interfaces. A new user needs to sign up with a valid email address, because our system will send notifications through emails. When the user logs in the system, he/she will be presented the profile page, which includes creating a new event, displaying the current ongoing events, reviewing past events which are terminated and a friend list. The user who creates the event severs as the admin of the event. The admin will need to provide event details as in Figure 2(a) including the category of the event, invited friends, the date and time of the event as well as its location. The system recommends 5 friends to users, which is based on the frequency of co-participated events. These details except friends are used to filter candidates. Friends invited and the admin define the group participating the event.

Based on this context the system generates a recommended list with 10 items tailored for this group as in Figure 2(b). Restaurants are ranked by the prediction of group score. The admin can select one or more options from our recommendation list or alternatively search yelp from our interface. Manual entries are also possible. After that, the admin will send out invitation to the group. Each participant in the group is called an attendee. The attendee will receive an email with a link to the voting page as in Figure 2(c).

We here describe how we designed these interfaces to meet the principle of consensus decision making [5].

Collaborative and Egalitarian. All members of the group contribute to a shared proposal and have the opportunity to present, and amend proposals. Thus, it is important that every member has the chance to propose restaurants to serve as options. The admin can select options during the event creation stage. All attendees including the admin can add more options on the voting page as in Figure 2(c). The interface for attendees adding new options is similar to the one in Figure 2(b). However, the recommended items will change if some users provided their feedback on some items already.

Inclusive and Participatory. In a consensus process all group members are included and encouraged to participate. In our system, every member will receive an email notification of the event and be asked to make decisions on options. The voting page does not require registration before access. We adopt the URL encoding method to create a special link for each user and each event. Users are uniquely identified by their emails. Non-registered users can also participate in the process. Whenever some members add more options, we will send an email to the group and ask them to provide feedback on the new options. Because members make decisions asynchronously, we provide a group chatting channel on the voting page. Members can leave messages and discuss on options.

Cooperative and Agreement Seeking. Participants in an effective consensus process should strive to reach the best possible decision for the group and all of its members, rather than competing for personal preferences. We display the number of positive and negative votes of each option in a real-time fashion. We clearly and immediately display the voting results in order to emphasize the goal of achieving high agreement on an option. The admin is the only person who can terminate the event and announce the final decision. When the admin terminated the event, every member will receive a summary of the event including the detail of options, votes and comments. The process and the result are transparent to every member, which avoids unfair bias to certain individuals.

We provide several types of information to each attendee to help them make decisions.

- Features of the item: below the title of each restaurant,

(a) Event Detail

(b) Recommendation

(c) Voting

(d) View Detail

Figure 2: Main Interfaces of the system

we embed a Yelp link. Users can click it and check the details. We also provide the address, Yelp rating, and the number of raters of the place if exist. Being presented first and close to a restaurant, the average rating and the number of Yelp raters are likely to make the strongest impression on users.

- Current votes: we display the accumulated positive votes and negative votes each option received so far. When a user provides his/her feedback on an option, the number of votes of this option changes correspondingly in real time.

- Members' choice: users can view who voted what on each option as in Figure 2(d). A person is affected by his friends' actions with different weight. Some members may have higher influence on a user than other members. Thus we disclose the voting detail and help users decide whose decision to follow.

Recall that our goal of the voting page is to provide an fair and convenient environment to do consensus decision making. In reality, when the group wanted to make such decisions, they may call each other and discuss the options, send many rounds of emails, or even organize a remote or face to face group meeting. All these efforts are to figure out what each member likes and dislikes and what options will be agreeable to all members. Through our system, we provide

a place that members can asynchronously discuss options, get to know other members' likes and dislikes, and conform to whoever is important to him/her. This information is collected and displayed in one page, and helps avoiding the burden of group discussion.

At last, the organizer needs to close the discussion and announce the final decision of the group by email notification. We rank options by the number of positive votes and populate the top one automatically. However, we still allow the organizer change the final decision to accommodate any possible change or offline discussion.

A system with new users will usually encounter the cold start problem. For this reason, our demo adopts content based algorithm. Every user has a history of voted restaurants. We extract features of each restaurant from Yelp, and predict the individual preference by its similarity with restaurants voted before. The group score is computed by the aggregation and the variance of individual predictions: $r(G,i) = w_1 \sum_{u \in G} r_{u,i} + w_2(1 - var(r_i))$. One restaurant may be voted many times in different events. The score of each restaurant is the running average of all votes it received by this user. This may cause the change of the predicted preference each time of a new vote.

3. SOCIAL INFLUENCE INFERENCE

In what follows, we discuss how we collect data from our system which can be used to improve the performance of recommendation. In particular, we propose a model that captures how a user is influenced by decisions made by other users. We demonstrate how we can learn parameters of the model from our data, and also show our model outperforms the baseline prediction using logistic regression.

Ye *et al.* [12] used a generative model to estimate the influence between two members. It assumes that a person first chooses friends with certain probability, then friends choose items. However, two users with similar interests consuming the same set of items will have high influence on each other. We argue that during the negotiation process, we can observe the influence between members not relying on the similarity of the interests. A user may vote for an item not because she likes that item but her friends like it a lot. Such influence will cause conformity within the group. We will describe the formal setting of the consensus negotiation process, then demonstrate how we leverage the feedback during this process to infer the social influence among members.

3.1 Decision Cascading Model

Consider the scenario that a group G with $|G| = d$ finite number of members tries to decide on the item i. During the consensus stage, every individual provides his/her own feedback to the item, which is represented by a binary value $y_{u,i} \in \{0, 1\}$. Note that $y_{u,i}$ can be multinomial variable such as "like", "dislike" or "neutral", and it can also be a real number scaling from 1 to 5. We here only deal with the binary case. Denote the group decision process by the group G to the item i by an event $e = (G, i, \{y_{u,i}\}_{u \in G})$. The decision of the item for the group depends on the number of positive votes. In other words, the average strategy is used to determine the group preference of an item as $y_{G,i} = 1/|G| \sum_{u \in G} y_{u,i}$.

We use $p(u|i)$ to denote the individual inherent preference on the item, $p(v|u)$ be the influence of the decision by the member v to the member u. We now model how each individual votes given he/she is aware of existing decisions of others. The order with which people vote can be arbitrary and depends on when they access the decision process. Moreover, their vote can be influenced by the votes of other people voted before them. Let $\mathcal{U}_e^+(u)$ be users who give positive feedback in event e before user u and $\mathcal{U}_e^-(u)$ be users who give negative feedback. The probability that the member u will give positive feedback to the item depends on his own preference on the item and other members' feedback as in (1). Putting differently, a user u votes positively by tossing $|\mathcal{U}_e^+(u)| + 1$ independent $0 - 1$ coins and observing whether any of them returning 1. The first coin is 1 with probability $p(u|i)$, and others with probability $p(v|u)$.

$$\Pr(y_{u,i} = 1) = 1 - (1 - p(u|i)) * \prod_{v \in \mathcal{U}_e^+(u)} (1 - p(v|u)) \quad (1)$$

3.2 Parameter Estimation

Given a set of events E, we can estimate the pairwise influence $p(v|u)$ and independent preference $p(u|i)$ by maximizing the likelihood of all events which the user u has participated.

$$L(E, u) = \prod_{e:y_{u,i}=1} \Pr(y_{u,i} = 1) \prod_{e:y_{u,i}=0} \Pr(y_{u,i} = 0) \quad (2)$$

We change the variables and let $1 - p(u|i) = q_{u,i}$, $r_{u,i} = \Pr(y_{u,i} = 1)$, $b_{v,u} = 1 - p(v|u)$. The problem becomes

$$\text{Maximize} \prod_{e:y_{u,i}=1} r_{u,i} \prod_{e:y_{u,i}=0} \prod_{v \in U_e^-(u)} b_{v,u} q_{u,i}$$

$$\text{subject to } 0 \leq r_{u,i} \leq 1; 0 \leq q_{u,i} \leq 1 \,\forall i$$

$$0 \leq b_{v,u} \leq 1 \,\forall v$$

$$r_{u,i} + \prod_{v \in U_e^+(u)} b_{v,i} q_{u,i} \leq 1 \,\forall e$$

Note the last constraint is an inequality rather than an equality. The objective function will strictly increase when either increase $r_{u,i}$ or $b_{v,i}$ or $q_{u,i}$, so the inequality will always be a binding constraint at the solution. The objective function is a monomial thus our problem is a geometric program which can be solved efficiently(see [10]).

The individual preference $p(u|i)$ reflects the interests of the individual and does not vary to a given event or a given group. We can estimate the individual preference from external resources, or by selecting events that the individual is the first one to make decision and using logistic regression on these events. Then $q_{u,i} = 1 - p(u|i)$ is treated as known constants when solving (2).

3.3 Evaluation

We use an offline evaluation to measure the performance of our decision cascading model. The basic idea is that if there is no pairwise influence on the decision, we can get good prediction on the individual feedback only using individual preference independently. Otherwise, our decision cascading model in (1) will achieve better prediction performance. We sampled 277 group events organized through our system. In total 19 individuals are involved, and 79 items are chosen. Groups have size 2,4 and 8. Every user participated in 54 events on average. For each event, members provide thumb up or thumb down feedback. A few of members did not provide any feedback. We collected 4398 valid votes at last. We split events chronologically into training(80%)and testing set (20%). The baseline algorithm uses logistic regression to predict individual preference.

	baseline	influence model
true positive rate	0.52	0.77
false positive rate	0.35	0.6
accuracy	0.65	0.7
AUC	0.61	0.85

Table 1: Prediction performance on test set, prediction threshold=0.5 for accuracy

Table 1 shows that the influence model has high true positive rate, better accuracy and much better AUC on the test set. However, the false positive rate is higher. This is caused that

in our system, we actually provided three options, since some participants usually do not click on the "Like" nor "Dislike". When we separately deal with negative votes, we can get similar performance improvement. In the future work, we can change the interface to collect only binary decisions. On the other hand, we can work on the model with multinomial decision values.

4. CONCLUSIONS

We designed a group recommender system which not only generates and presents recommendation suited for the group, but also facilitates the consensus decision making of the group. We collected events data through our system and observed feedback manifested in the consensus decision making process. We inferred the social influence within the group by our decision cascading model. The experiment results show that our model has better prediction performance. In the future, we will work on the recommendation algorithms using the social influence, and compare with existing algorithms.

5. REFERENCES

[1] S. Amer-Yahia, S. B. Roy, A. Chawlat, G. Das, and C. Yu. Group recommendation: semantics and efficiency. *Proc. VLDB Endow.*, 2(1):754–765, 2009.

[2] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso. Intrigue: Personalized recommendation of tourist attractions for desktop and handset devices. In *Applied Artificial Intelligence*, pages 687–714, 2003.

[3] L. Baltrunas, T. Makcinskas, and F. Ricci. Group recommendations with rank aggregation and collaborative filtering. In *Recsys*, 2010.

[4] M. O. Conner, D. Cosley, J. A. Konstan, and J. Riedl. Polylens: A recommender system for groups of users. In *Proceedings of the seventh conference on European Conference on Computer Supported Cooperative Work*, 2001.

[5] Consensusdecisionmaking.org. Consensus decision-making how to use consensus process. `http://www.consensusdecisionmaking.org/`, 08 2011.

[6] A. Crossen, J. Budzik, and K. J. Hammond. Flytrap: Intelligent group music recommendation. In *Intelligent User Interfaces*, 2002.

[7] L. H, D. NV, and V. A. LetÕs browse: a collaborative web browsing agent. In *Proceedings of the international conference on intelligent user interfaces*, 1999.

[8] J. Masthoff. Group modeling: selective sequence of television items. In *User modeling and user adapted interaction*, 2004.

[9] K. McCarthy, M. Salamo, L. Coyle, L. McGinty, B. Smyth, and P. Nixon. Group recommender systems: a critiquing based approach. In *IUI*, 2006.

[10] S. A. Myers and J. Leskovec. On the convexity of latent social network inference. In *NIPS*, 2010.

[11] G. Popescu and P. Pu. What's the best music you have? designing music recommendation for group enjoyment in groupfun. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, 2012.

[12] M. Ye, X. Liu, and W.-C. Lee. Exploring social influence for recommendation - a generative model approach. In *SIGIR*, 2012.

[13] Z. Yu, X. Zhou, Y. Hao, and J. Gu. Tv program recommendation for multiple viewers based on user profile merging. In *User Modeling and UserAdapted Interaction*, pages 63–82, 2006.