Probabilistic Game Theoretic Algorithms for Group Recommender Systems

George Popescu
EPFL - HCI Group
IC IIF Station 14
1015 Lausanne, Switzerland
+41 21 693 1246
george.popescu@epfl.ch

Pearl Pu
EPFL - HCI Group
IC IIF Station 14
1015 Lausanne, Switzerland
+41 21 693 6081
pearl.pu@epfl.ch

ABSTRACT
Aggregating users' individual preference and recommending a common set of items for a group has become a challenging topic in group recommender systems and social websites. This issue is mainly concerned with the following three objectives: eliciting individual users' preferences, suggesting outcomes that maximize the overall satisfaction for all users and ensuring that the aggregation mechanism is resistant to individual users' manipulation. Firstly we show how our proposed probabilistic weighted-sum algorithm (PWS) works and emphasize on its advantages. Then we compare PWS with related approaches implemented in similar systems using the case of our music recommender, GroupFun. We describe an experiment design to study users' perceptions of the algorithms, their perceived fairness and incentives to manipulate the final recommendation outcome. We expect our results to show that PWS will be perceived as fair and diversity- and discovery-driven, thus enhancing the group's satisfaction. Our future work will focus on the actual evaluation of GroupFun using the experiment design presented here.

Categories and Subject Descriptors
H1.2 [**User/Machine Systems**]: human factors; H5.2 [**User Interfaces**]: evaluation/methodology, user-centered design.

General terms
Experimentation, Human factors.

Keywords
Quality measurement, usability evaluation, recommender systems, quality of user experience, post-study questionnaire.

1. INTRODUCTION
Group recommender systems use various aggregation strategies to suggest a common list of items to a group of users. These strategies aim at increasing the group's welfare and are based on users' votes on items. The social welfare is an aggregate of individual utilities of all group members. Most common used deterministic strategies are: plurality voting, utilitarian, approval voting, least misery, most pleasure, average without misery, fairness, most respected person, Borda count, Copeland rule or Kemeny scores (Masthoff, 2005). One can easily create other distinct strategies based on these. Social choice theory aims to offer an answer to "which strategy is most effective and will be most liked by a group of users?" (Hastie and Kameda, 2005). With the purpose of determining what strategy people actually use, Masthoff (2004) found that individuals use computationally simple strategies mentioned above, particularly the average strategy, the average without misery and the least misery strategy. However, there is no dominant strategy as people switch between them given a different context. Fairness plays an important role in decision making but members do not have a clear strategy for applying it.

Our main research question is to determine "which group satisfaction rule best satisfy users expectations". We propose 4 algorithms and investigate upon: "which algorithm is best suited to meet users' expectations" for our music recommender system, GroupFun and "how users perceive the algorithms' accuracy". Next we present related work, then considered algorithms together with our implementation and future experiment design.

WOMRAD 2011 2ndWorkshop on Music Recommendation and Discovery, colocated with ACM RecSys 2011 (Chicago, US)

2. BASELINE AND RELATED WORK

2.1 MusicFX
MusicFX is a music system offering best music match to employees working out in a fitness center (McCarthy and Anagnost, 1998). The algorithm aims at selecting the most preferred music genre that maximizes members' listening pleasure. For this it computes a group preference index and sums squared individual preferences. Then it lists the most popular categories. The system also saves events in its history such as: member entrance, member exit, individual preference update, system parameter adjustment and maximum station play time elapsed. Since some individual preference filters may not change, the system opts for a different music configuration according to two criteria: playing more the music which members like most and playing less the music which members like least. The weighted random selection operator is one strategy used to reduce the likelihood of starvation. Another strategy is limiting the period of time for one genre to be played – regardless of how popular it is – before the selection algorithm is invoked in order to select a new station. MusicFX has two important advantages: (1) it increases the variety of music and (2) it democratizes the music selection process. Thus it is adaptive to changing preferences of its users also proposing new songs for them. One drawback of the system is that it changes music stations abruptly in the middle of the songs.

2.2 PolyLens
PolyLens is a collaborative filtering recommender system which recommends movies to groups of people based on their individual preferences (O'Connor et al. 2001). It represents a group

extension of the MovieLens recommender system with over 80,000 users and their ratings for more than 3,500 movies (with a total of nearly 5 million votes). Users can create and manage groups, access individual and group recommendations and receive notification alerts for group invitation. The algorithm uses the least misery strategy given that the groups formed to watch a movie together tend to be small. As such the group is as happy as its least happy member. The authors mention that "the social value function and algorithm are unlikely to work well for large groups". They further note that "it is still an open research question to understand the types of social value functions that best satisfy large groups and to implement them algorithmically".

2.3 Voting strategies

An extensive study on a group of television viewers aiming at finding which strategy people use was realized by Masthoff (2005). In the experiment 10 deterministic group voting rules are compared: plurality voting, utilitarian strategy, Borda count, Copeland rule, approval voting, least misery strategy, most pleasure strategy, average without misery strategy, fairness strategy and most respected person strategy. The experiment shows that individuals do not use a clear dominant strategy, but average, average without misery, and least misery are all plausible candidates for implementation. In a different experiment addressing how people judge the recommendation results multiplicative utilitarian strategy is the most promising strategy, but the other strategies received close scores. In the study of television viewers the hypothesis that social status influences selection has no statistical dominance. Non-linear utility suits better users' expectations than a linear one. For instance quadratic rating scale is appropriate for implementation. Furthermore, there is strong evidence that human subjects use a series of simple strategies in different judgment contexts they face. For instance, if one takes the satisfaction of the group to be the average of the satisfaction of the individuals, then the average strategy performs best. Taking the minimum better corresponds to the predictions which are made by individuals assessing their own needs.

3. ALGORITHMS

In the music domain many users usually form many groups and listen to many songs. Given the fact that the length of one song is of 3 to 4 minutes users usually select a playlist containing several to lots of songs. This is not the case of movies selection when users need to agree on only one or few movie(s) they would like to consume given their limited time and the length of a movie: ~2h. Thus, the music domain presents both opportunities and challenges since the recommendation needs to focus on both diversity and accuracy.

We propose the following 4 algorithms for comparison:

- PS (Probabilistic Sum): select the common playlist' songs probabilistically, each of them having the same probability to be selected
- LM (Least Misery): select songs with the highest minimum individual ratings
- DWS (Deterministic Weighted Sum): deterministically select songs with the highest score
- PWS (Probabilistic Weighted Sum): compute weighted sum and select songs based on their score probabilities.

3.1 General framework

Let A be the set of all users and S the set of all possible outcomes that can be rated. In our group music recommendation setting, the outcomes are songs s_i that are selected in the common playlist. We let each user a_j submit a numerical vote $score(s_i, a_j)$ for each song s_i that reflects their preference for that song. These votes are given as ratings on a 5-point Likert scale and normalized so that the scores given by each user sum to 1:

$$score(s_i, a_j) = \frac{rating(s_i, a_j)}{\sum_i rating(s_i, a_j)} \tag{1}$$

We then assign a joint score to each song that is computed as the sum of the scores given by the individual users:

$$score(s_i) = \sum_{a_j \in A} score(s_i, a_j) \tag{2}$$

To choose the songs to be included in a playlist of length k, a deterministic method is to choose the k songs with the highest joint rating: weighted sum (DWS):

$$score(s_i) = \frac{score(s_i)}{\sum_{s_i \in S} score(s_i)} \tag{3}$$

The probabilistic weighted sum (PWS) iteratively selects each of the k songs randomly according to the probability distribution:

$$p(s_i) = \frac{score(s_i)}{\sum_{s_i \in S} score(s_i)} \tag{4}$$

By comparison, the probabilistic sum (PS) method chooses the k songs with equal probability:

$$p(s_i) = \frac{1}{|S|} \tag{5}$$

The least misery (LM) method takes into account the minimum of ratings for each user:

$$\min\left(score(s_i, a_j)\right), \forall a_j \in A \tag{6}$$

3.2 Example

To illustrate how each algorithm works, we consider the following example. In the next table, user1, user2, and user 3 represent group members. The score distribution normalized to 1 for each of the users is displayed in the respective row, and the joint scores are shown in the table below.

Table I. Item selection example using the 4 algorithms

User1	Song1: 0.1	Song2: 0.4	Song3: 0.4	Song4: 0.1
User2	Song1: __	Song2: 0.2	Song3: __	Song4: 0.8
User3	Song1: 0.4	Song2: 0.2	Song3: __	Song4: 0.4
Total	Song1: 0.5	Song2: 0.8	Song3: 0.4	Song4: 1.3

The least misery (LM) will choose song 2 and song 3 (each of them has the minimal rating 0.2). For all other songs the minimum score is 0.1. After normalizing the total scores by the sum of scores, we obtain the following probability distribution for the set of outcomes.

Table II. Probability distribution

P	Song1: 0.16	Song2: 0.26	Song3: 0.13	Song 4: 0.43

Considering the probability as the final score, the deterministic weighted sum (DWS) will chose songs 4, 2, 1 and 3. Probabilistic weighted sum (PWS) will choose one song after another using this probabilistic distribution. Compared to other social choice based algorithms, PWS is incentive compatible. That is, it is to the best interest of the individual to reveal his/her preferences truthfully. It is in fact equivalent to a random dictator method, where the dictator will choose a song randomly with the probabilities given by its degree of preference – a reasonable method since nobody wants to hear the same song over and over again. This is because the probability of a song s_i to be chosen can be written as:

$$p(s_i) = \frac{score(s_i)}{|A|} = \sum_{a_j \in A} \frac{score(s_i, a_j)}{|A|} \qquad (7)$$

or, in other words, the probability of choosing user a_j times the normalized score that user a_j has given to song s_i. Indeed, User3's preference for song 1 yields a significant probability that this song will be included in the playlist, relative to other songs.

3.3 Discussion

The contribution of the PWS algorithm in the paper stands out with respect to group satisfaction. We expect users to be more satisfied using PWS than other algorithms given their expectations to discover the music of other members.

Advantages of PWS compared with the other algorithms:

1. Users are free to choose the number of songs
2. Ratings are updated permanently
3. The algorithm is computationally simple
4. Users can negotiate their ratings and trade utility
5. Incentive-compatible truthful property is observed
6. The algorithm favors music diversity

The disadvantages of PWS are:

1. It is difficult to quantify rating differences between distinct users. The weights given by each user cannot be compared with the ones given by another since users have different estimations of their utility.
2. Self-selection effect: most popular songs will receive most votes - not ideal if long tail distribution is desired.

Since PWS can be interpreted as similar to the random scheme users have to test it in more recommendation rounds to understand its inner logic. PWS can be further developed to include the group dynamics. One solution is to consider trust and other members' comments on the songs rated by one user (e.g. "like"/"dislike").

The PWS algorithm stands out with respect to allowing users to engage in trustful individual preference elicitation and music discovery. By returning to the recommendation list the group will find a different playlist every-time they are would like to listen to group music. By considering the probabilistic distribution of ratings and an extensive music library the algorithm will mostly suggest songs strongly liked by most others. Sometimes it will recommend unexpected, rating-wise, serendipitous items facilitating music discovery and group enjoyment.

4. GROUPFUN

GroupFun is a web application that helps a group of friends to agree on a common music playlist for a given event they will attend, e.g. a birthday party or a graduation ceremony. Firstly, it is implemented as a Facebook plugin connecting users to their friends. Secondly, it is a music application that helps individuals to manage and share their favorite music with groups. In GroupFun users can listen to their own collection of songs as well as their friends' music. With the collective music database, the application integrates friends' music tastes and recommends a common playlists to them. Therefore, the application aims at satisfying music tastes of the whole group by aggregating individual preferences through the use of previously presented algorithms.

Figure 1. "Home" page of GroupFun

In the "Home" page users see 4 playlists: one from a recent event, one containing popular songs, one from a party and the last one from an older event. They can listen to each song in each of the playlists.

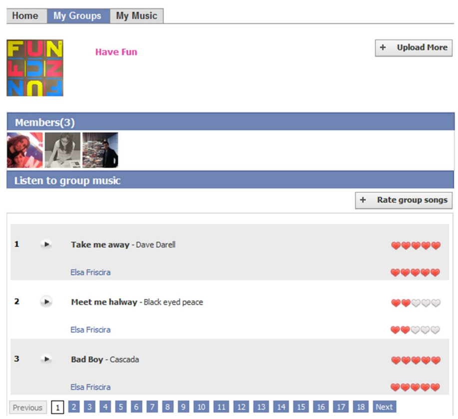

Figure 2. "My Groups" page of GroupFun

In the "My Group" page users can create groups, upload and rate their music, invite friends and hear the group's songs. Finally, in the "My Music" page users see their contribution to GroupFun: for each song is displayed the associated group, the rating and its name and artist. Users can also listen to their individual preferences in the same interface. One of the most important characteristics of GroupFun is that it combines music, friends and groups together. In other words, GroupFun serves as a platform allowing users to conveniently organize their individual music library, effectively communicate with friends and actively participate in social activities.

5. EXPERIMENT DESIGN

To compare how users perceive the 4 algorithms, we plan to carry on a between-groups user study. With the results of the experiment we will be able to make a judgment of the influence of both the algorithms and the design on users' satisfaction. We plan to collect solid user feedback regarding how an algorithm should allow group members to arrive at a common decision in a music recommendation setting.

Our hypotheses are that: (1) users will not reveal their preferences strategically as to influence the algorithm's outcome; (2) they will prefer more PWS than DWS given the increased diversity of the recommendations they receive and (3) the group will perceive more overall satisfaction but less diversity using the LM algorithm compared with PWS.

5.1 Procedure

First we recruit university students, friends who have Facebook accounts and other users on the Amazon Mechanical Turk platform. All of them will use their own computers in order to connect to the GroupFun application. We consider 4 algorithms implemented for 4 groups of 40 users together with 1 interface. Each of the algorithms displays a common list of 10 songs to all group members. Users are asked to contribute their music to only one of the groups and fill in an online post-study questionnaire assessing their satisfaction. The final music outcome is shown to all group members after they have finished their tasks. Users can interact with the system in diverse ways such as: upload more or less songs, change their ratings, see and hear to their friends' playlists, etc.

Table III. Evaluation of 4 algorithms using the same interface

Interface/Algorithm	**PWS**	**DWS**	**LM**	**PS**
Interface	40	40	40	40

5.2 Measurements

The first 40 users see the results of the probabilistic weighted sum algorithm, the next those of deterministic weighted sum and so on. Since individuals who upload more songs expect to see their song names more often in the final list they would prefer to know a priori the computation rule of the algorithm so that they would adjust the number of songs they upload. Given users' known self-ratings and group ratings computed by the algorithms we expected our subjects to identify some differences between the 4 approaches. Some of the questions from the post-study questionnaire are presented in the table below. They were extracted from a well-known user evaluation model, named ResQue [7], that our group has developed.

Table IV. Evaluation questions

Measurements	*Questions*
Perceived attractiveness	The layout of the system's interface is attractive.
Perceived satisfaction	The items recommended to me matched my interest.
Perceived helpfulness	I took into account the ratings given by my friends.
Outcome change intention	I was interested in changing the outcome of the algorithm.

6. CONCLUSIONS AND FUTURE WORK

In this paper we presented our research work on the algorithmic development and evaluation of our music recommender system and Facebook application, GroupFun. The major contribution of this paper is the demonstration of the applicability of the PWS algorithm for group recommendation strategies and negotiation. In this context, we analyzed different group recommendation approaches w.r.t. group satisfaction and discussed key satisfaction issues to be taken into account. The PWS algorithm we proposed calculates probabilities for songs to appear in groups' playlists favoring music diversity and discovery. Using PWS users state their preference truthfully. They align their decision to that of the group. Furthermore, our current development of GroupFun allows users to create groups, rate and share their music profiles with their friends.

To understand how users' perceive our algorithms and current interface, we plan to conduct an experiment to compare the 4 algorithms in a between-subjects study. As such we will evaluate user satisfaction for music group recommendations. Furthermore, to learn more about the perceived ease of use and perceived usefulness of our system we plan to invite more members and analyze user feedback in terms of design and functionality. We also intend to develop a new version of the algorithm which will better match users' behavior and expectations.

7. ACKNOWLEDGMENTS

This research was supported by EPFL and the Swiss National Science Foundation. We thank Elsa Friscira and Laurentiu Dascalu for their important contribution to the development of GroupFun and GroupFun users for their answers and involvement in the evaluation of our algorithms.

8. REFERENCES

[1] Hastie, R. and Kameda, T. 2005. The robust beauty of majority rules in group decisions. In *Psychological Review*, Vol. 112, no. 2, 494-508.

[2] Masthoff, J. 2004. Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Modeling and User-Adapted Interaction*, Vol. 14, no. 1, 37-85.

[3] Masthoff, J. 2005. The pursuit of satisfaction: affective state in group recommender systems. In Computer Science, Vol. 3538, 297-306.

[4] McCarthy J.F. and Anagnost, T.D. 1998. MusicFX: an arbiter of group preferences for computer supported collaborative workouts. *Proc. ACM Computer Supported Cooperative Work*, 363-372.

[5] O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J. 2001. PolyLens: A recommender system for groups of users. *Proc. ACM European Conference on Computer Supported Cooperative Work*.

[6] Popescu G. and Pu P. 2010. Group Recommender Systems as a Voting Problem. *EPFL Technical report*.

[7] Pu P. and Chen L. 2010. A User-Centric Evaluation Framework of Recommender Systems. In the *3rd ACM Conference on Recommender Systems*, Barcelona, Spain