Group Recommender Systems: Combining Individual Models

Chapter · January 2011

DOI: 10.1007/978-0-387-85820-3_21

Judith Masthoff
Utrecht University

Group Recommender Systems: Combining individual models

Judith Masthoff

Abstract This chapter shows how a system can recommend to a group of users by aggregating information from individual user models and modelling the users affective state. It summarizes results from previous research in this area. It also shows how group recommendation techniques can be applied when recommending to individuals, in particular for solving the cold-start problem and dealing with multiple criteria.

1 Introduction

Most work on recommender systems to date focuses on recommending items to individual users. For instance, they may select a book for a particular user to read based on a model of that user's preferences in the past. The challenge recommender system designers traditionally faced is how to decide what would be optimal for an individual user. A lot of progress has been made on this, as evidenced by other chapters in this handbook (e.g. [3, 8, 9, 14, 15]).

In this chapter, we go one-step further. There are many situations when it would be good if we could recommend to a group of users rather than to an individual. For instance, a recommender system may select television programmes for a group to view or a sequence of songs to listen to, based on models of all group members. Recommending to groups is even more complicated than recommending to individuals. Assuming that we know perfectly what is good for individual users, the issue arises how to combine individual user models. In this chapter, we will discuss how group recommendation works, what its problems are, and what advances have been made. Interestingly, we will show that group recommendation techniques have many uses as well when recommending to individuals. So, even if you are developing recom-

Judith Masthoff
University of Aberdeen, AB24 3UE Aberdeen UK, e-mail: j.masthoff@abdn.ac.uk

mender systems aimed at individual users you may still want to read on (perhaps reading Section 7 first will convince you).

This chapter focusses on deciding what to recommend to a group, in particular how to aggregate individual user models. There are other issues to consider when building a group recommender system which are outside the scope of this chapter. In particular:

- *How to acquire information about individual users' preferences.* The usual recommender techniques can be used (such as explicit ratings and collaborative- and content-based filtering, see other handbook chapters). There is a complication in that it is difficult to infer an individual's preferences when a group uses the system, but inferences can be made during individual use combined with a probabilistic model when using it in company. An additional complication is that an individual's ratings may depend on the group they are in. For instance, a teenager may be very happy to watch a programme with his younger siblings, but may not want to see it when with his friends.
- *How will the system know who is present?* Different solutions exist, such as users explicitly logging in, probabilistic mechanisms using the time of day to predict who is present, the use of tokens and tags, etc [17].
- *How to present and explain group recommendations?* As seen in this handbook's chapter on explanations, there are already many considerations when presenting and explaining *individual* recommendations. The case of group recommendations is even more difficult. More discussion on explaining group recommendations is provided in [13] and under Challenges in our final section.
- *How to help users to settle on a final decision?* In some group recommenders, users are given group recommendations, and based on these recommendations negotiate what to do. In other group recommenders this is not an issue (see Section 2.3 on the difference between passive and active groups). An overview of how users' decisions can be aided is provided in [13].

The next section highlights usage scenarios of group recommenders, and provides a classification of group recommenders inspired by differences between the scenarios. Section 3 discusses strategies for aggregating models of individual users to allow for group recommendation, what strategies have been used in existing systems, and what we have learned from our experiments in this area. Section 4 deals with the issue of order when we want to recommend a sequence of items. Section 5 provides an introduction into the modelling of affective state, including how an individual's affective state can be influenced by the affective states of other group members. Section 6 explores how such a model of affective state can be used to build more sophisticated aggregation strategies. Section 7 shows how group modelling and group recommendation techniques can be used when recommending to an individual user. Section 8 concludes this chapter and discusses future challenges.

2 Usage Scenarios and Classification of Group Recommenders

There are many circumstances in which adaptation to a group is needed rather than to an individual. Below, we present two scenarios that inspired our own work in this area, discuss the scenarios underlying related work, and provide a classification of group recommenders inspired by differences between the scenarios.

2.1 Interactive Television

Interactive television offers the possibility of personalized viewing experiences. For instance, instead of everybody watching the same news program, it could be personalized to the viewer. For me, this could mean adding more stories about the Netherlands (where I come from), China (a country that fascinates me after having spent some holidays there) and football, but removing stories about cricket (a sport I hardly understand) and local crime. Similarly, music programs could be adapted to show music clips that I actually like.

There are two main differences between traditional recommendation as it applies to say PC-based software and the interactive TV scenarios sketched above. Firstly, in contrast to the use of PCs, television viewing is largely a family or social activity. So, instead of adapting the news to an individual viewer, the television would have to adapt it to the group of people sitting in front of it at that time. Secondly, traditional work on recommendation has often concerned recommending one particular thing to the user, so for instance, which movie the user should watch. In the scenarios sketched above, the television needs to adapt a sequence of items (news items, music clips) to the viewer. The combination of recommending to a group and recommending a sequence is very interesting, as it may allow you to keep all individuals in the group satisfied by compensating for items a particular user dislikes with other items in the sequence which they do like.

2.2 Ambient Intelligence

Ambient intelligence deals with designing physical environments that are sensitive and responsive to the presence of people. For instance, consider the case of a bookstore where sensors detect the presence of customers identified by some portable device (e.g. a Bluetooth-enabled mobile phone, or a fidelity card equipped with an active RFID tag). In this scenario, there are various sensors distributed among the shelves and sections of the bookstore which are able to detect the presence of individual customers. The bookstore can associate the identification of customers with their profiling information, such as preferences, buying patterns and so on.

With this infrastructure in place, the bookstore can provide customers with a responsive environment that would adapt to maximise their well-being with a view to increasing sales. For instance, the device playing the background music should take into account the preferences of the group of customers within hearing distance. Similarly, LCD displays scattered in the store show recommended books based on the customers nearby, the lights on the shop's display window (showing new titles) can be rearranged to reflect the preferences and interests of the group of customers watching it, and so on. Clearly, group adaptation is needed, as most physical environments will be used by multiple people at the same time.

2.3 Scenarios Underlying Related Work

In this section we discuss the scenarios underlying the best known group recommender systems:

- MUSICFX [22] chooses a radio station for background music in a fitness centre, to suit a group of people working out at a given time. This is similar to the Ambient Intelligence scenario discussed above.
- POLYLENS [25] is a group recommender extension of MOVIELENS. MOVIELENS recommends movies based on an individual's taste as inferred from ratings and social filtering. POLYLENS allows users to create groups and ask for group recommendations.
- INTRIGUE [4] recommends places to visit for tourist groups taking into account characteristics of subgroups within that group (such as children and the disabled).
- The TRAVEL DECISION FORUM [12] helps a group to agree on the desired attributes of a planned joint holiday. Users indicate their preferences on a set of features (like sport and room facilities). For each feature, the system aggregates the individual preferences, and users interact with embodied conversational agents representing other group members to reach an accepted group preference.
- The COLLABORATIVE ADVISORY TRAVEL SYSTEM (CATS) [23] also helps users to choose a joint holiday. Users consider holiday packages, and critique their features (e.g., 'like the one shown but with a swimming pool'). Based on these critiques, the system recommends other holidays to them. Users also select holidays they like for other group members to see, and these are annotated with how well they match the preferences of each group member (as induced from their critiques). The individual members' critiques results in a group preference model, and other holidays are recommended based on this model.
- YU'S TV RECOMMENDER [29] recommends a television program for a group to watch. It bases its recommendation on the individuals' preferences for program features (such as genre, actors, keywords).

2.4 A Classification of Group Recommenders

The scenarios provided above differ on several dimensions, which provide a way to classify group recommender systems:

- *Individual preferences are known versus developed over time.* In most scenarios, the group recommender starts with individual preferences. In contrast, in CATS, individual preferences develop over time, using a critiquing style approach. Another chapter discusses critiquing and its role in group recommendation [24].
- *Recommended items are experienced by the group versus presented as options*. In the Interactive TV scenario, the group experiences the news items. In the Ambient Intelligence and MUSICFX scenarios, they experience the music. In contrast, in the other scenarios, they are presented with a list of recommendations. For example, POLYLENS presents a list of movies the group may want to watch.
- *The group is passive versus active.* In most scenarios, the group does not interact with the way individual preferences are aggregated. However, in the TRAVEL DECISION FORUM and CATS the group negotiates the group model.
- *Recommending a single item versus a sequence.* In the scenarios of MUSICFX, POLYLENS, and YU'S TV RECOMMENDER it is sufficient to recommend individual items: people normally only see one movie per evening, radio stations can play forever, and YU'S TV RECOMMENDER chooses one TV program only. Similarly, in the TRAVEL DECISION FORUM and CATS users only go on one holiday. In contrast, in our Interactive TV scenario, a sequence of items is recommended, for example making up a complete news broadcast. Similarly, in INTRIGUE, it is quite likely that a tourist group would visit multiple attractions during their trip, so would be interested in a sequence of attractions to visit. Also, in the Ambient Environment scenario it is likely that a user will hear multiple songs, or see multiple items on in-store displays.

In this chapter, we will focus on the case where individual preferences are known, the group directly experiences the items, the group is passive, and a sequence is recommended. Recommending a sequence raises interesting questions regarding sequence order (see Section 4) and considering the individuals' affective state (see Sections 5 and 6). A passive group with direct experience of the items makes it even more important that the group recommendation is good.

DeCampos et al.'s classification of group recommenders also distinguishes between passive and active groups [7]. In addition, it uses two other dimensions:

- *How individual preferences are obtained.* They distinguish between content-based and collaborative filtering. Of the systems mentioned above, POLYLENS is the only one that uses collaborative filtering.
- *Whether recommendations or profiles are aggregated.* In the first case, recommendations are produced for individuals and then aggregated into a group recommendation. In the second case, individual preferences are aggregated into a group model, and this model is used to produce a group recommendation. They mention INTRIGUE and POLYLENS as aggregating recommendations, while the others aggregate profiles.

These two dimensions are related to how the group recommender is implemented rather than being inherent to the usage scenario. In this chapter, we focus on aggregating profiles, but the same aggregation strategies apply when aggregating recommendations. The material presented in this chapter is independent of how the individual preferences are obtained.

3 Aggregation Strategies

The main problem group recommendation needs to solve is how to adapt to the group as a whole based on information about individual users' likes and dislikes. For instance, suppose the group contains three people: Peter, Jane and Mary. Suppose a system is aware that these three individuals are present and knows their interest in each of a set of items (e.g. music clips or advertisements). Table 1 gives example ratings on a scale of 1 (really hate) to 10 (really like). Which items should the system recommend, given time for four items?

Table 1 Example of individual ratings for ten items (A to J)

	A	B	C	D	E	F	G	H	I	J
Peter	10	4	3	6	10	9	6	8	10	8
Jane	1	9	8	9	7	9	6	9	3	8
Mary	10	5	2	7	9	8	5	6	7	6

3.1 Overview of Aggregation Strategies

Many strategies exist for aggregating individual ratings into a group rating (e.g. used in elections and when selecting a party leader). For example, the Least Misery Strategy uses the minimum of ratings to avoid misery for group members (Table 2).

Table 2 Example of the Least Misery Strategy

	A	B	C	D	E	F	G	H	I	J
Peter	10	4	3	6	10	9	6	8	10	8
Jane	1	9	8	9	7	9	6	9	3	8
Mary	10	5	2	7	9	8	5	6	7	6
Group Rating	1	4	2	6	7	8	5	6	3	6

Eleven aggregation strategies inspired by Social Choice Theory are summarised in Table 3 (see [17] for more details).

Table 3 Overview of Aggregation Strategies

Strategy	How it works	Example
Plurality Voting	Uses 'first past the post': repetitively, the item with the most votes is chosen.	A is chosen first, as it has the highest rating for the majority of the group, followed by E (which has the highest rating for the majority when excluding A).
Average	Averages individual ratings	B's group rating is 6, namely (4+9+5)/3.
Multiplicative	Multiplies individual ratings	B's group rating is 180, namely 4*9*5.
Borda Count	Counts points from items' rankings in the individuals' preference lists, with bottom item getting 0 points, next one up getting one point, etc	A's group rating is 17, namely 0 (last for Jane) + 9 (first for Mary) + 8 (shared top 3 for Peter)
Copeland Rule	Counts how often an item beats other items (using majority vote) minus how often it looses	F's group rating is 5, as F beats 7 items (B,C,D,G,H,I,J) and looses from 2 (A,E).
Approval Voting	Counts the individuals with ratings for the item above a approval threshold (e.g. 6)	B's group rating is 1 and F's is 3.
Least Misery	Takes the minimum of individual ratings	B's group rating is 4, namely the smallest of 4,9,5.
Most Pleasure	Takes the maximum of individual ratings	B's group rating is 9, namely the largest of 4,9,5.
Average without Misery	Averages individual ratings, after excluding items with individual ratings below a certain threshold (say 4).	J's group rating is 7.3 (the average of 8,8,6), while A is excluded because Jane hates it.
Fairness	Items are ranked as if individuals are choosing them in turn.	Item E may be chosen first (highest for Peter), followed by F (highest for Jane) and A (highest for Mary).
Most respected person	Uses the rating of the most respected individual.	If Jane is the most respected person, then A's group rating is 1. If Mary is most respected, then it is 10.

3.2 Aggregation Strategies Used in Related Work

Most of the related work uses one the aggregation strategies in Table 3 (sometimes with a small variation), and they differ in the one used:

- INTRIGUE uses a weighted form of the Average strategy. It bases its group recommendations on the preferences of subgroups, such as children and the disabled. It takes the average, with weights depending on the number of people in the subgroup and the subgroup's relevance (children and disabled were given a higher relevance).
- POLYLENS uses the Least Misery Strategy, assuming groups of people going to watch a movie together tend to be small and that a small group tends to be as happy as its least happy member.
- MUSICFX uses a variant of the Average Without Misery Strategy. Users rate all radio stations, from +2 (really love this music) to -2 (really hate this music). These ratings are converted to positive numbers (by adding 2) and then squared to widen the gap between popular and less popular stations. An Average Without Misery strategy is used to generate a group list: the average of ratings is taken but only for those items with individual ratings all above a threshold. To avoid starvation and always picking the same station, a weighted random selection is made from the top stations of the list.
- YU'S TV RECOMMENDER uses a variant of the Average Strategy. It bases its group recommendation on individuals' ratings of program features: -1 (dislikes the feature), +1 (likes the feature) and 0 (neutral). The feature vector for the group minimizes its distance compared to individual members' feature vectors. This is similar to taking the average rating per feature.
- The TRAVEL DECISION FORUM has implemented multiple strategies, including the Average Strategy and the Median Strategy. The Median strategy (not in Table 3) uses the middle value of the ratings. So, in our example, this results in group ratings of 10 for A, and 9 for F. The Median Strategy was chosen because it is nonmanipulable: users cannot steer the outcome to their advantage by deliberately giving extreme ratings that do not truly reflect their opinions. In contrast, for example, with the Least Misery strategy devious users can avoid getting items they dislike slightly, by giving extremely negative ratings. The issue of manipulability is most relevant when users provide explicit ratings, used for group recommendation only, and are aware of others' ratings, all of which is the case in the TRAVEL DECISION FORUM. It is less relevant when ratings are inferred from user behaviour, also used for individual recommendations, and users are unaware of the ratings of others (or even of the aggregation strategy used).
- In CATS, users indicate through critiquing which features a holiday needs to have. For certain features, users indicate whether they are required (e.g. ice skating required). For others, they indicate quantities (e.g. at least 3 ski lifts required). The group model contains the requirements of all users, and the item which fulfils most requirements is recommended. Users can also completely discard holidays, so, the strategy has a Without Misery aspect.

It should be noted that both YU'S TV RECOMMENDER and the TRAVEL DECISION FORUM aggregate preferences for each feature without using the idea of fairness: loosing out on one feature is not compensated by getting your way on another.

Though some exploratory evaluation of MUSICFX, POLYLENS and CATS has taken place, for none of these systems it has been investigated how effective their strategy really is, and what the effect would be of using a different strategy. The experiments presented in the next section shed some light on this question.

In contrast, some evaluation of YU'S TV RECOMMENDER has taken place [29]. They found that their aggregation worked well when the group was quite homogenous, but that results were disliked when the group was quite heterogeneous. This is as we would expect, given the Average Strategy will make individuals quite happy if they are quite similar, but will cause misery when tastes differ widely.

3.3 Which Strategy Performs Best

We conducted a series of experiments to investigate which strategy from Table 3 is best (see [17] for details).

In Experiment 1 (see Figure 1), we investigated how people would solve this problem, using the User as Wizard evaluation method [20]. Participants were given individual ratings identical to those in Table 1. These ratings were chosen to be able to distinguish between strategies. Participants were asked which items the group should watch, if there was time for one, two, .., seven items. We compared participants' decisions and rationale with those of the aggregation strategies. We found that participants cared about fairness, and about preventing misery and starvation ("this one is for Mary, as she has had nothing she liked so far"). Participants' behaviour reflected that of several of the strategies (e.g. the Average, Least Misery, and Average Without Misery were used), while other strategies (e.g. Borda count, Copeland rule) were clearly not used.

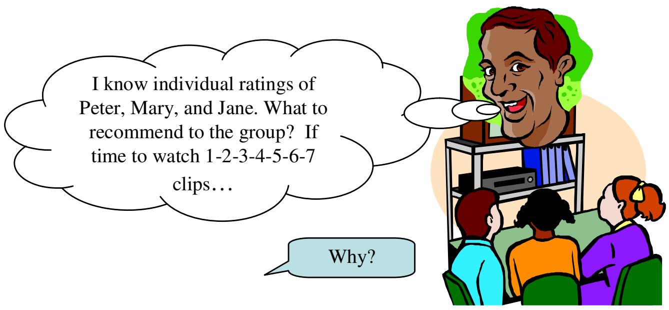

Fig. 1 Experiment 1: which sequence of items do people select if given the system's task

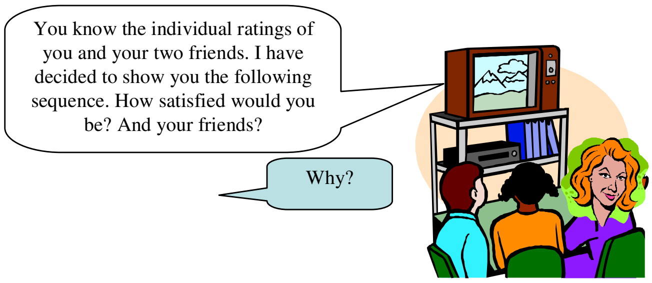

Fig. 2 Experiment 2: What do people like?

In Experiment 2 (see Figure 2), participants were given item sequences chosen by the aggregation strategies as well as the individual ratings in Table 1. They rated how satisfied they thought the group members would be with those sequences, and explained their ratings. We found that the Multiplicative Strategy (which multiplies the individual ratings) performed best, in the sense that it was the only strategy for which *all* participants thought its sequence would keep all members of the group satisfied. Borda count, Average, Average without Misery and Most Pleasure also performed quite well. Several strategies (such as Copeland rule, Plurality voting, Least misery) could be discarded as they clearly were judged to result in misery for group members.

We also compared the participants' judgements with predictions by simple satisfaction modelling functions. Amongst other, we found that more accurate predictions resulted from using:

- quadratic ratings, which e.g. makes the difference between a rating of 9 and 10 bigger than that between a rating of 5 and 6
- normalization, which takes into account that people rate in different ways, e.g., some always use the extremes of a scale, while others only use the middle of the scale.

4 Impact of Sequence Order

As mentioned in Section 2, we are particularly interested in recommending a *sequence* of items. For example, for a personalised news program on TV, a recommender may select seven news items to be shown to the group. To select the items, it can use an aggregation strategy (such as the Multiplicative Strategy) to combine individual preferences, and then select the seven items with the highest group ratings. Once the items have been selected, the question arises in what order to show them in the news program. For example, it could show the items in descending order of group rating, starting with the highest rated item and ending with the lowest rated one. Or, it could mix up the items, showing them in a random order.

However, the problem is actually far more complicated than that. Firstly, in responsive environments, the group membership changes continuously, so deciding on the next seven items to show based on the current members seems not a sensible strategy, as in the worse case, none of these members may be present anymore when the seventh item is shown.

Secondly, overall satisfaction with a sequence may depend more on the order of the items than one would expect. For example, for optimal satisfaction, we may need to ensure that our news program has:

- *A good narrative flow.* It may be best to show topically related items together. For example, if we have two news items about Michael Jackson (say about his funeral and about a tribute tour) then it seems best if these items are presented together. Similarly, it would make sense to present all sports' items together.
- *Mood consistency.* It may be best to show items with similar moods together. For example, viewers may not like seeing a sad item (such as a soldier's death) in the middle of two happy items (such as a decrease in unemployment and a sporting victory).
- *A strong ending.* It may be best to end with a well-liked item, as viewers may remember the end of the sequence most.

Similar ordering issues arise in other recommendation domains. For example, a music programme may want to consider rhythm when sequencing items. The recommender may need additional information (such as items' mood, topics, rhythm) to optimise ordering. It is beyond the topic of this chapter to discuss how this can be done (and is very recommender domain specific). We just want to highlight that the items already shown may well influence what the best next item is. For example, suppose the top four songs in a music recommender were all Blues. It may well be that another Blues song ranked sixth may be a better next selection than a Classical Opera song ranked fifth.

In Experiment 3 (see Figure 3), we investigated how a previous item may influence the impact of the next item. Amongst others, we found that mood (resulting from the previous item) and topical relatedness can influence ratings for subsequent items. This means that aggregating individual profiles into a group profile should be done repeatedly, every time a decision needs to be made about the next item to display.

5 Modelling Affective State

When recommending to a group of people, you cannot give everybody what they like all of the time. However, you do not want anybody to get too dissatisfied. For instance, in a shop it would be bad if a customer were to leave and never come back, because they really cannot stand the background music. Many shops currently opt to play music that nobody really hates, but most people not love either. This may prevent loosing customers, but would not result in increasing sales. An ideal shop

Fig. 3 Experiment 3: Investigating the effect of mood and topic

would adapt the music to the customers in hearing range in such a way that they get songs they really like most of the time (increasing the likelihood of sales and returns to the shop). To achieve this, it is unavoidable that customers will occasionally get songs they hate, but this should happen at a moment when they can cope with it (e.g. when being in a good mood because they loved the previous songs). Therefore, it is important to monitor continuously how satisfied each group member is. Of course, it would put an unacceptable burden on the customers if they had to rate their satisfaction (on music, advertisements etc) all the time. Similarly, measuring this satisfaction via sensors (such as heart rate monitors or facial expression recognizers) is not yet an option, as they tend to be too intrusive, inaccurate or expensive. So, we propose to model group members' satisfaction; predicting it based on what we know about their likes and dislikes.

5.1 Modelling an Individual's Satisfaction on its Own

In [19], we investigated four satisfaction functions to model an individual's satisfaction. We compared the predictions of these satisfaction functions with the predictions of real users. We also performed an experiment (see Figure 4) to compare the predictions with the real feelings of users.

The satisfaction function that performed best defines the satisfaction of a user with a new item *i* after having seen a sequence *items* of items as:

$$Sat(items + < i >) = \frac{\delta \times Sat(items) + Impact(i, \delta \times Sat(items))}{1+\delta}$$

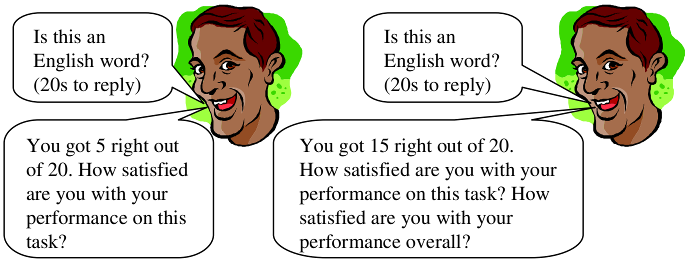

Fig. 4 Experiment 4: Measuring overall satisfaction during a series of tasks

with the impact on satisfaction of new item i given existing satisfaction s defined as

$$Impact(i,s) = Impact(i) + (s - Impact(i)) \times \varepsilon, \text{ for } 0 \leq \varepsilon \leq 1 \text{ and } 0 \leq \delta \leq 1$$

Parameter δ represents satisfaction decaying over time (with δ=0 past items have no influence, with δ=1 there is no decay).

Parameter ε represents the influence of the user's satisfaction after experiencing previous items on the impact of a new item. This parameter is inspired by the psychology and economics literature, which shows that mood impacts evaluative judgement [19]. For instance, half the participants answering a questionnaire about their TVs received a small present first to put them in a good mood. These participants were found to have televisions that performed better. So, if a user is in a good mood due to liking previous items, the impact of an item they normally dislike may be smaller (with how much smaller depending on ε).

Parameters δ and ε are user dependent (as confirmed in the experiment in [19]). We will not define $Impact(i)$ in this chapter, see [19] for details, but it involves quadratic ratings and normalization as found in the experiment discussed above.

5.2 Effects of the Group on an Individual's Satisfaction

The satisfaction function given does not take the satisfaction of other users in the group into account, which may well influence a user's satisfaction. As argued in [19] based on social psychology, two main processes can take place.

Emotional Contagion. Firstly, the satisfaction of other users can lead to so-called emotional contagion: other users being satisfied may increase a user's satisfaction (e.g. if somebody smiles at you, you may automatically smile back and feel better as a result). The opposite may also happen: other users being dissatisfied may decrease a user's satisfaction. For instance, if you are watching a film with a group of friends than the fact that your friends are clearly not enjoying it may negatively impact your own satisfaction.

Emotional contagion may depend on your personality (some people are more easily contaged than others), and your relationship with the other person. Anthropologists and social psychologists have found substantial evidence for the existence of four basic types of relationships, see Figure 5. In Experiment 5 (see Figure 6), we confirmed that emotional contagion indeed depends on the relationship you have: you are more likely to be contaged by somebody you love (like your best friend) or respect (like your mother or boss) then by somebody you are on equal footing with or are in competition with.

Conformity. Secondly, the opinion of other users may influence your own expressed opinion, based on the so-called process of conformity.

Figure 7 shows the famous conformity experiment by Asch [5]. Participants were given a very easy task to do, like decide which of the four lines has the same orientation as the line in Card A. They thought they were surrounded by other participants, but in fact the others where part of the experiment team. The others all answered the question before them, picking the same wrong answer. It was shown that most participants then pick that same wrong answer as well.

Two types of conformity exist: (1) normative influence, in which you want to be part of the group and express an opinion like the rest of the group even though inside you still belief differently, and (2) informational influence, in which your own opin-

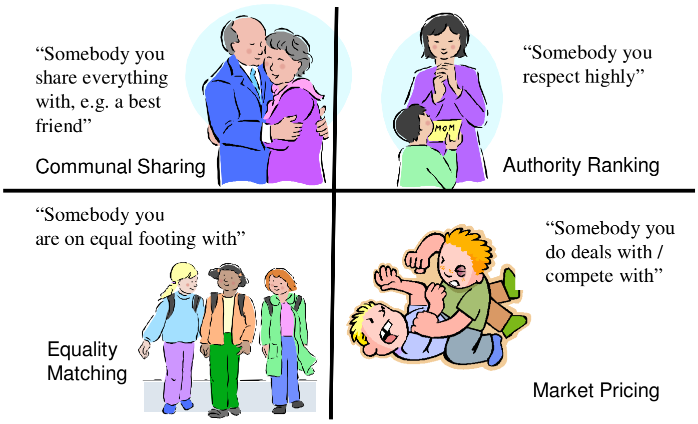

Fig. 5 Types of relationship

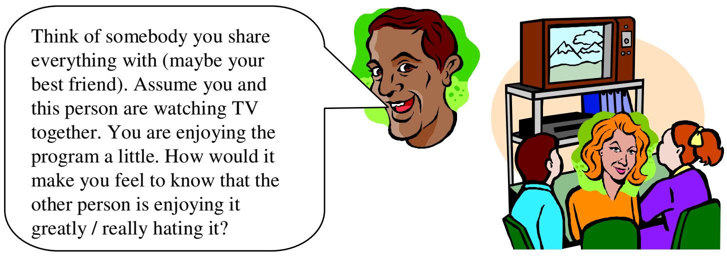

Fig. 6 Experiment 5: Impact of relationship type on emotional contagion

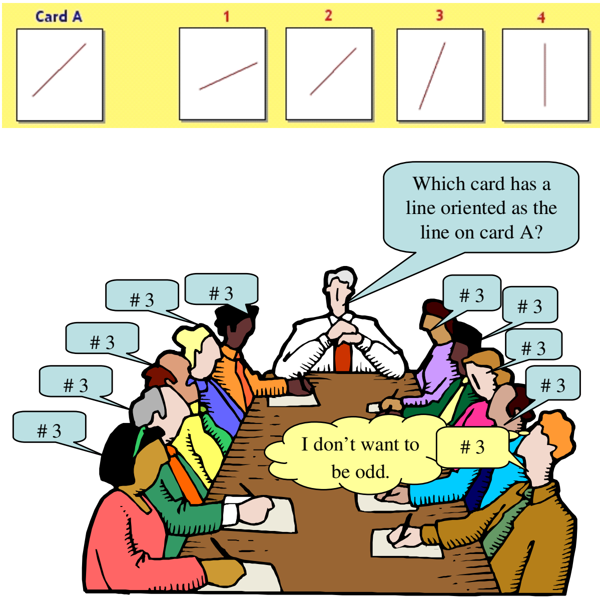

Fig. 7 Conformity experiment by Asch

ion changes because you believe the group must be right. Informational influence would change your own satisfaction, while normative influence can change the satisfaction of others through emotional contagion because of the (insincere) emotions you are portraying.

More complicated satisfaction functions are presented in [19] to model emotional contagion and both types of conformity.

6 Using Affective State inside Aggregation Strategies

Once you have an accurate model of the individual users' satisfaction, it would be nice to use this model to improve on the group aggregation strategies. For instance, the aggregation strategy could set out to please the least satisfied member of the group. This can be done in many different ways, and we have only started to explore this issue. For example:

- *Strongly Support Grumpiest strategy.* This strategy picks the item which is *most liked* by the least satisfied member. If multiple of these items exist, it uses one of the standard aggregation strategies, for instance the Multiplicative Strategy, to distinguish between them.
- *Weakly Support Grumpiest strategy.* This strategy selects the items that are *quite liked* by the least satisfied member, for instance items with a rating of 8 or above.

It uses one of the standard aggregation strategies, like the Multiplicative Strategy, to choose between these items.

- *Weighted strategy.* This strategy assign weights to users depending on their satisfaction, and then use a weighted form of a standard aggregation strategy. For instance, Table 4 shows the effect of assigning double the weight to Jane when using the Average Strategy. Note that weights are impossible to apply to a strategy like the Least Misery Strategy.

Table 4 Results of Average strategy with equal weights and with twice the weight for Jane

	A	B	C	D	E	F	G	H	I	J
Peter	10	4	3	6	10	9	6	8	10	8
Jane	1	9	8	9	7	9	6	9	3	8
Mary	10	5	2	7	9	8	5	6	7	6
Average (equal weights)	7	6	4.3	7.3	8.7	8.7	5.7	7.7	6.7	7.3
Average (Jane twice)	5.5	6.8	5.3	8.3	8.3	8.8	5.8	8	5.8	7.5

In [21], we discuss this in more detail, propose an agent-based architecture for applying these ideas to the ambient intelligent scenario, and describe an implemented prototype. Clearly, empirical research is needed to investigate the best way of using affective state inside an aggregation strategy.

7 Applying Group Recommendation to Individual Users

So, what if you are developing an application that recommends to a single user? Group recommendation techniques can be useful in three ways: (1) to aggregate multiple criteria, (2) to solve the so-called cold-start problem, (3) to take into account opinions of others. The chapter on aggregation of preferences also discusses how aggregation may be needed when recommending to individuals, and covers several specific aggregation functions [6].

7.1 Multiple Criteria

Sometimes it is difficult to give recommendations because the problem is multidimensional: multiple criteria play a role. For instance, in a news recommender system, a user may have a preference for location (being more interested in stories close to home, or related to their favourite holiday place). The user may also prefer more recent news, and have topical preferences (e.g. preferring news about politics to news about sport). The recommender system may end up with a situation like in Table 5, where different news story rate differently on the criteria. Which news stories should it now recommend?

Table 5 Ratings on criteria for 10 news items

	A	B	C	D	E	F	G	H	I	J
Topic	10	4	3	6	10	9	6	8	10	8
Location	1	9	8	9	7	9	6	9	3	8
Recency	10	5	2	7	9	8	5	6	7	6

Table 5 resembles the one we had for group recommendation above (Table 1), except that now instead of multiple users we have multiple criteria to satisfy. It is possible to apply our group recommendation techniques to this problem. However, there is an important difference between adapting to a group of people and adapting to a group of criteria. When adapting to a group of people, it seems sensible and morally correct to treat everybody equally. Of course, there may be some exceptions, for instance when the group contains adults as well as children, or when it is somebody's birthday. But in general, equality seems a good choice, and this was used in the group adaptation strategies discussed above. In contrast, when adapting to a group of criteria, there is no particular reason for assuming all criteria are as important. It is even quite likely that not all criteria are equally important to a particular person. Indeed, in an experiment we found that users treat criteria in different ways, giving more importance to some criteria (e.g. recency is seen as more important than location) [18]. So, how can we adapt the group recommendation strategies to deal with this? There are several ways in which this can be done:

- Apply the strategy to the most respected criteria only. The ratings of unimportant criteria are ignored completely. For instance, assume criterion Location is regarded unimportant, then its ratings are ignored. Table 6 shows the result of the Average Strategy when ignoring Location.
- Apply the strategy to all criteria but use weights. The ratings of unimportant criteria are given less weight. For instance, in the Average Strategy, the weight of a criterion is multiplied with its ratings to produce new ratings. For instance, suppose criteria Topic and Recency were three times as important as criterion Location. Table 7 shows the result of the Average Strategy using these weights.
  In case of the Multiplicative Strategy, multiplying the ratings with weights does not have any effect. In that strategy, it is better to use the weights as exponents, so replace the ratings by the ratings to the power of the weight. Note that in both strategies, a weight of 0 results in ignoring the ratings completely, as above.
- Adapt a strategy to behave differently to important versus unimportant criteria: Unequal Average Without Misery. Misery is avoided for important criteria but not for unimportant ones. Assume criterion Location is again regarded as unimportant. Table 8 shows the results of the Unequal Average Without Misery strategy with threshold 6.

Table 6 Average Strategy ignoring unimportant criterion Location

	A	B	C	D	E	F	G	H	I	J
Topic	10	4	3	6	10	9	6	8	10	8
Recency	10	5	2	7	9	8	5	6	7	6
Group	20	9	5	13	19	17	11	14	17	14

Table 7 Average Strategy with weights 3 for Topic and Recency and 1 for Location

	A	B	C	D	E	F	G	H	I	J
Topic	30	12	9	18	30	27	18	24	30	24
Location	1	9	8	9	7	9	6	9	3	8
Recency	30	15	6	21	27	24	15	18	21	18
Group	61	36	23	48	64	60	39	51	54	50

Table 8 Unequal Average Without Misery Strategy with Location unimportant and threshold 6

	A	B	C	D	E	F	G	H	I	J
Topic	10	4	3	6	10	9	6	8	10	8
Location	1	9	8	9	7	9	6	9	3	8
Recency	10	5	2	7	9	8	5	6	7	6
Group	21			22	26	26		23	20	22

We have some evidence that people's behaviour reflects the outcomes of these strategies [18], however, more research is clearly needed in this area to see which strategy is best. Also, more research is needed to establish when to regard a criterion as ”unimportant”. The issue of multiple criteria is also the topic of another chapter in this handbook [1].

7.2 Cold-Start Problem

A big problem for recommender systems is the so-called cold-start problem: to adapt to a user, the system needs to know what the user liked in the past. This is needed in content-based filtering to decide on items similar to the ones the user liked. It is needed in social filtering to decide on the users who resemble this user in the sense that they (dis)liked the same items in the past (see Figure 8). So, what if you do not

Fig. 8 Cold-start problem in case of social-filtering

know anything about the user yet, because they only just started using the system? Recommender system designers tend to solve this problem by either getting users to rate items at the start, or by getting them to answer some demographic questions (and then using stereotypes as a starting point, e.g. elderly people like classical music).

Both methods require user effort. It is also not easy to decide which items to get a user to rate, and stereotypes can be quite wrong and offensive (some elderly people prefer pop music and people might not like being classified as elderly).

The group recommendation work presented in this chapter provides an alternative solution. When a user is new to the system, we simply provide recommendations to that new user that would keep the whole group of existing users happy. We assume that our user will resemble one of our existing users, though we do not know which one, and that by recommending something that would keep all of them happy, the new user will be happy as well.

Gradually, we will learn about the new user's tastes, for instance, by them rating our recommended items or, more implicitly, by them spending time on the items or not. We provide recommendations to the new user that would keep the group of existing users happy including the new user (or more precisely, the person we now assume the new user to be). The weight attached to the new user will be low initially, as we do not know much about them yet, and will gradually increase. We also start to attach less weight to existing users whose taste now evidently differs from our new user.

Figure 9 shows an example of the adaptation: the system is including the observed tastes of the new user to some extent, and has started to reduce the weights of some of the other users. After prolonged use of the system, the user's inferred wishes will completely dominate the selection.

We have done a small-scale study using the MovieLens dataset to explore the effectiveness of this approach. We randomly selected five movies, and twelve users who had rated them: ten users as already known to the recommender, and two as new users. Using the Multiplicative Strategy on the group of known users, movies were ranked for the new users. Results were encouraging: the movie ranked highest was in fact the most preferred movie for the new users, and also the rest of the

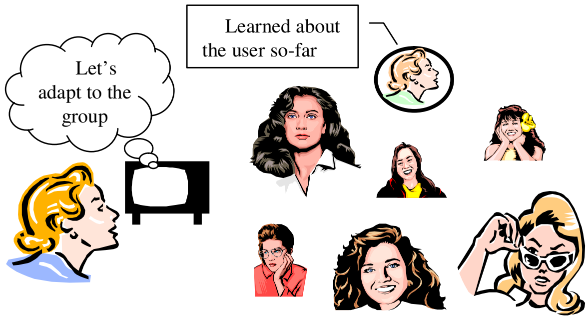

Fig. 9 Gradually learning about the user, and whom she resembles most

ranking was fine given the new users' profiles. Applying weights led to a further improvement of the ranking, and weights started to reflect the similarity of the new users with known users. More detail on the study and on applying group adaptation to solve the cold-start problem is given in [16].

7.3 Virtual Group Members

Finally, group adaptation can also be used when adapting to an individual by adding virtual members to the group. For instance, a parent may be fine with the television entertaining their child, but may also want the child occasionally to learn something. When the child is alone, the profile of the parent can be added to the group as a virtual group member, and the TV could try to satisfy both.

8 Conclusions and Challenges

Group recommendation is a relatively new research area. This chapter is intended as an introduction in the area, in particular on aggregating individual user profiles. For more detail please see [17, 19, 21, 18, 16, 12, 13].

8.1 Main Issues Raised

The main issues raised in this chapter are:

- Adapting to groups is needed in many scenarios such as interactive TV, ambient intelligence, recommending to tourist groups, etc. Inspired by the differences

between scenarios, group recommenders can be classified using multiple dimensions.
- Many strategies exist for aggregating individual preferences (see Table 3), and some perform better than others. Users seem to care about avoiding misery and fairness.
- Existing group recommenders differ on the classification dimensions and in the aggregation strategies used. See Table 9 for an overview.
- When recommending a sequence of items, aggregation of individual profiles has to occur at each step in the sequence, as earlier items may impact the ratings of later items.
- It is possible to construct satisfaction functions to predict how satisfied an individual will be at any time during a sequence. However, group interaction effects (such as emotional contagion and conformity) can make this complicated.
- It is possible to evaluate in experiments how good aggregation strategies and satisfaction functions are, though this is not an easy problem.
- Group aggregation strategies are not only important when recommending to groups of people, but can also be applied when recommending to individuals, e.g. to prevent the cold-start problem and deal with multiple criteria.

8.2 Caveat: Group Modelling

The term ”group modelling” is also used for work that is quite different from that presented in this chapter. A lot of work has been on modelling common knowledge between group members (e.g. [11, 27], modelling how a group interacts (e.g. [26, 10]) and group formation based on individual models (e.g. [26, 2]).

8.3 Challenges

Compared to work on individual recommendations, group recommendation is still quite a novel area. The work presented in this chapter is only a starting point. There are many challenging directions for further research, including:

- *Recommending item sequences to a group.* Our own work seems to be the only work to date on recommending balanced *sequences* that address the issue of fairness. Even though sequences are important for the usage scenario of INTRIGUE, their work has not investigated making sequences balanced nor has it looked at sequence order. Clearly, a lot more research is needed on recommending and ordering sequences, in particular on how already shown items should influence the ratings of other items. Some of this research will have to be recommender domain specific.
- *Modelling of affective state.* There is a lot more work needed to produce validated satisfaction functions. The work presented in this chapter and [19] is only

the starting point. In particular, large scale evaluations are required, as are investigations on the affect of group size.

- *Incorporating affective state within an aggregation strategy* As noted in Section 6, there are many ways in which affective state can be used inside an aggregation strategy. We presented some initial ideas in this area, but extensive empirical research is required to investigate this further.
- *Explaining group recommendations: Transparency and Privacy* One might think that accurate predictions of individual satisfaction can also be used to improve the recommender's transparency: showing how satisfied other group members are could improve users' understanding of the recommendation process and perhaps make it easier to accept items they do not like. However, users' need for privacy is likely to conflict with their need for transparency. An important task of a group recommender system is to avoid embarrassment. Users often like to conform to the group to avoid being disliked (we discussed normative conformity as part of Section 5.2 on how others in the group can influence an individual's affective state). In [19], we have investigated how different group aggregation strategies may affect privacy. More work is needed on explanations of group recommendations, in particular on how to balance privacy with transparency and scrutability. The chapter on explanations provides more detail on the different roles of explanations in recommender systems [28].
- *User interface design.* An individual's satisfaction with a group recommendation may be increased by good user interface design. For example, when showing an item, users could be shown what the next item will be (e.g. in a TV programme through a subtitle). This may inform users who do not like the current item that they will like the next one better.
- *Group aggregation strategies for cold-start problems*. In Section 7.2, we have sketched how group aggregation can be used to help solve the cold-start problem. However, our study in this area was very small, and a lot more work is required to validate and optimise this approach.
- *Dealing with uncertainty.* In this chapter, we have assumed that we have accurate profiles of individuals' preferences. For example, in Table 1, the recommender knows that Peter's rating of item B is 4. However, in reality we will often have probabilistic data. For example, we may know with 80% certainty that Peter's rating is 4. Adaptations of the aggregation strategies may be needed to deal with this. DeCampos et al try to deal with uncertainty by using Baysian networks [7]. However, they have so far focussed on the Average and Plurality Voting strategies, not yet tackling the avoidance of misery and fairness issues.
- *Empirical Studies*. More empirical evaluations are vital to bring this field forwards. It is a challenge to design well-controlled, large scale empirical studies in a real-world setting, particularly when dealing with group recommendations and affective state. All research so far (including my own) has either been on a small scale, in a contrived setting or lacks control.

Table 9 Group recommender systems

System	Usage scenario	Classification				Strategy used
		Preferences known	Direct Experience	Group Active	Recommends Sequence	
MUSICFX [22]	Chooses radio station in fitness centre based on people working out	Yes	Yes	No	No	Average Without Misery
POLYLENS [25]	Proposes movies for a group to view	Yes	No	No	No	Least Misery
INTRIGUE [4]	Proposes tourist attractions to visit for a group based on characteristics of subgroups (such as children and the disabled)	Yes	No	No	Yes	Average
TRAVEL DECISION FORUM [12]	Proposes a group model of desired attributes of a planned joint vacation and helps a group of users to agree on these	Yes	No	Yes	No	Median
YU'S TV RECOMMENDER [29]	Proposes a TV program for a group to watch based on individuals' ratings for multiple features	Yes	No	No	No	Average
CATS [23]	Helps users to choose a joint holiday, based on individuals' critiques of proposed holidays	No	No	Yes	No	Counts requirements met Uses Without Misery
MASTHOFF'S GROUP RECOMMENDER [17, 19]	Chooses a sequence of music video clips for a group to watch based on individuals' ratings	Yes	Yes	No	Yes	Multiplicative etc

Acknowledgements Judith Masthoff's research has been partly supported by Nuffield Foundation Grant No. NAL/00258/G.

References

1. Adomavicius, G., Manouselis, N., Kwon, Y.: Multi-criteria recommender systems. In this handbook (2010).
2. Alfonseca, E., Carro, R.M., Martn, E., Ortigosa, A., Paredes, P.: The Impact of Learning Styles on Student Grouping for Collaborative Learning: A Case Study. UMUAI 16 (2006) 377-401
3. Amatriain, X., Jaimes, A., Oliver, N., Puyol, J.M.: Data mining methods for recommender systems. In this handbook (2010).
4. Ardissono, L., Goy,A., Petrone, G., Segnan, M., Torasso, P.: Tailoring the Recommendation of Tourist Information to Heterogeneous User Groups. In S. Reich, M. Tzagarakis, P. De Bra (eds.), Hypermedia: Openness, Structural Awareness, and Adaptivity, International Workshops OHS-7, SC-3, and AH-3. Lecture Notes in Computer Science 2266, Springer Verlag, Berlin (2002) 280-295
5. Asch,S.E.:Studies of independence and conformity: a minority of one against a unanimous majority. Pschol. Monogr. 70 (1956) 1-70
6. Beliakov, G., Calvo, T., James, S.: Aggregation of preferences in recommender systems. In this handbook (2010).
7. de Campos, L.M., Fernandez-Luna, J.M., Huete, J.F., Rueda-Morales, M.A.: Managing uncertainty in group recommending processes. UMUAI 19 (2009) 207-242
8. Desrosiers, C., Karypis, G.: A comprehensive study of neighborhood-based recommendation methods. In this handbook (2010).
9. Felfernig, A., Friedrich, G., Jannach, D., Zanker, M.: Developing contraint-based recommenders. In this handbook (2010).
10. Harrer, A., McLaren, B.M., Walker, E., Bollen L., Sewall, J.: Creating Cognitive Tutors for Collaborative Learning: Steps Toward Realization. UMUAI 16 (2006) 175-209
11. Introne, J., Alterman,R.: Using Shared Representations to Improve Coordination and Intent Inference. UMUAI 16 (2006) 249-280
12. Jameson, A.: More than the Sum of its Members: Challenges for Group Recommender Systems. International Working Conference on Advanced Visual Interfaces, Gallipoli, Italy (2004)
13. Jameson, A., Smyth, B.: Recommendation to groups. In: Brusilovsky, P., Kobsa, A., Njedl, W. (Eds). The Adaptive Web Methods and Strategies of Web Personalization. Springer (2007) 596-627
14. Koren, Y., Bell, R.: Advanced in collaborative filtering. In this handbook (2010).
15. Lops, P., de Gemmis, M., Semeraro, G.: Content-based recommender systems: State of the art and trends. In this handbook (2010).
16. Masthoff, J.: Modeling the multiple people that are me. In: P. Brusilovsky, A.Corbett, and F. de Rosis (eds.) Proceedings of the 2003 User Modeling Conference, Johnstown, PA. Springer Verlag, Berlin (2003) 258-262
17. Masthoff, J.: Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers. UMUAI 14 (2004) 37-85
18. Masthoff, J.: Selecting News to Suit a Group of Criteria: An Exploration. 4th Workshop on Personalization in Future TV - Methods, Technologies, Applications for Personalized TV, Eindhoven, the Netherlands (2004)
19. Masthoff, J., Gatt, A.: In Pursuit of Satisfaction and the Prevention of Embarrassment: Affective state in Group Recommender Systems. UMUAI 16 (2006) 281-319

20. Masthoff, J.: The user as wizard: A method for early involvement in the design and evaluation of adaptive systems. Fifth Workshop on User-Centred Design and Evaluation of Adaptive Systems (2006).
21. Masthoff, J., Vasconcelos, W.W., Aitken, C., Correa da Silva, F.S.: Agent-Based Group Modelling for Ambient Intelligence. AISB symposium on Affective Smart Environments, Newcastle, UK (2007)
22. McCarthy, J., Anagnost, T.: MusicFX: An Arbiter of Group Preferences for Computer Supported Collaborative Workouts. CSCW, Seattle, WA. (1998) 363-372
23. McCarthy, K., McGinty, L., Smyth, B., Salamo, M.: The needs of the many: A case-based group recommender system. European Conference on Case-Based Reasoning, Springer (2006) 196-210
24. McGinty, L., Reilly, J. On the evolution of critiquing recommenders. In this handbook (2010).
25. O' Conner, M., Cosley, D., Konstan, J.A., Riedl, J.: PolyLens: A Recommender System for Groups of Users. ECSCW, Bonn, Germany (2001) 199-218. As accessed on http://www.cs.umn.edu/Research/GroupLens/poly-camera-final.pdf
26. Read, T., Barros, B., Brcena, E., Pancorbo, J.: Coalescing Individual and Collaborative Learning to Model User Linguistic Competences. UMUAI 16 (2006) 349-376
27. Suebnukarn, S., Haddawy, P.: Modeling Individual and Collaborative Problem-Solving in Medical Problem-Based Learning. UMUAI 16 (2006) 211-248
28. Tintarev, N., Masthoff, J.: Guidelines for designing and evaluating explanations for recommender systems. In this handbook (2010).
29. Yu, Z., Zhou, X., Hao, Y. Gu, J.: TV Program Recommendation for Multiple Viewers Based on User Profile Merging. UMUAI 16 (2006) 63-82