

User Satisfaction in Long Term Group Recommendations*

Lara Quijano-Sánchez, Juan A. Recio-García, and Belén Díaz-Agudo

Dep. Ingeniería del Software e Inteligencia Artificial,
Universidad Complutense de Madrid, Spain
{lara.quijano,jareciog}@fdi.ucm.es, belend@sip.ucm.es

Abstract. In this paper we introduce our application *HappyMovie*, a Facebook application for movie recommendation to groups. This system takes advantage of social data available in this social network to promote fairness for the provided recommendations. Group recommendations are based in the individual satisfaction of each individual. The (in)satisfaction of users modifies the typical aggregation functions used to estimate the value of an item for the group. This paper proposes a memory of past recommendations to compute the satisfaction of users when similar items (movies, in this case) are recommended several times.

1 Introduction

Recommender systems were born from the necessity of having some kind of guidance when searching through complex product spaces. More precisely, group recommenders were built to help groups of people, who share a common activity, decide in conflict situations. Our previous works [1,2,3,4] presented our approach of making recommendations for groups of people connected through social network structures. In them we introduced a method, based on three important features: personality, social trust and memory of past recommendations to ensure social fairness. We have proven that our theories for making recommendations to groups of people connected through social network structures improve the current existing methods.

Our current work consists on providing an Facebook application reachable to a great deal of people, where we can continue our research and experiments and also extend the group recommendations to different domains. Besides, by having the application located in a social network, we can extract the information regarding the users from it. In that way we don't have to bombard our users with questionnaires in order to build the personal profile that we need to do the recommendation, because we can extract from their Facebook profiles a great deal of the information we need. We have moved our standalone group recommendation application to a public application where everybody can benefit of it. Our method includes the analysis of the group personality composition and the trust

* Supported by the Spanish Ministry of Science and Education (TIN2009-13692-C03-03).

between users to improve the accuracy of group recommenders. This way we simulate in a more realistic way the argumentation process followed by groups of people when agreeing on a common activity. Our recommendation method and the architecture of our Facebook application are valid for any domains with rated products. However, we have applied them to an specific domain, the movie domain, and we have created *Happy Movie*.

The main goal of this paper is to analyse the impact of the last social factor involved in our group recommendation method: group satisfaction to ensure social fairness. The use of group satisfaction is based on some results from organizational behaviour and social psychology that have highlighted the concept of *emotional contagion* [5]. This social aspect states that the satisfaction of an individual is likely to depend on other individuals of the group [6,7]. In this context, social fairness is understood as the intention of maximizing the personal satisfaction of every user in the group and minimizing their differences. To achieve this goal, we propose a memory of past recommendations that is used to compute an individual satisfaction value that is later combined to estimate the global satisfaction of the group for the provided recommendations. This group satisfaction is critical in recommender systems that propose the same kind of items several times for the same group of people. Some examples are movies, music, leisure trips and any other social activity.

Several works have focused in avoiding repeated recommendations and recommendations that tend to be detrimental for the same group members repeatedly. MusicFX [8] employs a weighted random policy for selecting one of the top radio stations selected by the recommender, instead of always selecting the top category. Another solution is taking into account the history of the results produced by the recommender. For example, in FlyTrap [9] the previous selections are taken in consideration. This way, when they choose the next song to be played, abrupt changes of genre do not appear. Another system that takes into account the previous selections is PoolCasting [10]. It uses a Case-Based Reasoning system to generate a sequence of songs customized for a community of listeners. To select each song in the sequence, first a subset of songs musically associated with the last song of the sequence is retrieved from a music pool; then the preferences of the audience expressed as cases are reused to customize the selection for the group of listeners; finally, listeners can revise their satisfaction (or lack of) for the songs they have heard.

The process that we have followed when making group recommendations which ensure social fairness is very similar to the Case Base Reasoning (CBR) cycle [11]. CBR is a successful and established methodology in Artificial Intelligence that has inspired us in the implementation of our recommender with memory. In our system each recommendation provided by the group recommender will be stored as a new case that can be used later to improve the next recommendation. This fact corresponds with the *retain phase* in the CBR cycle. This way, we acquire the experiences that will be useful for the resolution of future recommendations because we will know which products have been recommended to a group. We also store how much satisfied each member of the group

is with this recommendation, so we are able to adjust the satisfaction factor in future recommendations. Before making the following recommendation we will check the previous situation, which in the CBR cycle will be the *retrieve phase*. Once we obtain that information we can perform a new recommendation, but taking into account what we have retained (the products that we have already recommended and how satisfied each of the members of the group are). This will be equivalent to the *reuse phase* in the CBR cycle. Last but not least, we will modify the recommendation so that the proposed products are not repeated and we assure a certain degree of fairness when we benefit the preferences of each users. This last phase, the *revise* one, will close de CBR cycle.

The next section details our recommender application. In Section 3 we explain the recommendation techniques used by the application to select items for the group. Experimental evaluation is exposed in Section 4 and finally conclusions are detailed in Section 5.

2 Facebook Application: Happy Movie

HappyMovie is a Facebook application where we provide a group recommendation for groups of people that wish to go together to the cinema. Although this application has been initially designed for recommending movies, this domain will be extended as our method is valid for any other domain with rated products.

In the following sections we are going to explain the uses of the application and its architecture.

2.1 Uses of Happy Movie

In order to use our application, users only have to start their Facebook account and look for *HappyMovie* in the applications section. We explain the uses of the application through an example of a given group of people connected in the social network. The necessary steps to obtain a movie recommendation for a group with Happy movie are explained below:

- **Creating the user profile in the application:** Before any user has access to the movie recommendation results we have to create their individual “recommendation profile” which is necessary for our recommendation method. This profile is based on three different aspects: personality, individual preferences and trust with other users. To obtain the personality and preferences, users must answer two different tests. The first one is the personality test, where users have to choose a series of characters to whom they feel identified, as shown in Figure 1 (left image). Once they have answered the personality test, users have to rate a set of movies (at least 20 movies), where they suggest their personal preferences , as shown in Figure 1 (right image). Finally, to obtain the last factor –trust– the application reads the information stored in the Facebook personal profile. It calculates the trust that the user has with all the other users that have joined the event up to now.



Fig. 1. Personality and Preferences test in *HappyMovie*

- **Creating the activity:** The organizer user, U_1 decides to organize an activity and starts the application to create a new event “*Let’s go to the cinema*” as shown in Figure 2 (left picture). To create the event, organizers must establish some data like place, date or invited users. Once the event has been created any user attending the event can see the date and place of the event and a proposal of three movies, that are the best ones that our group recommender has found for the current group of users attending the event. For example, when U_1 starts the application for the first time she has the role of *organizer*. As an *organizer* she firstly invites some of her Facebook friends to the event. Lets say that she invites U_2 and U_3 . Next, she chooses the place and date where the event will take place.

Once the *organizer* finalizes this initial configuration she will continue with a role of *common user*. When users participate in the event as *common users* they are also able to invite any Facebook friends they wish and they can retire from the event at any time. For example, U_2 accepts the invitation of the event and later she invites her Facebook friend U_4 . On the other hand, U_3 initially accepts the invitation and joins the event but later she decides against going, so she retires from it.

- **Recommendation method:** When the application has obtained the three factors that identify each user that joins the event (personality, individual preferences and trust between other users) it provides a group recommendation using our concrete method which is explained in Section 3.
- **Having the recommendation made:** When the event is created it looks up for the current movie listing from the selected city and provides a list of 3 movies, which represent the best 3 movies that the recommender has found in the movie listing for the users that have joined the event up to now, this is shown in Figure 2 (right picture). This list is automatically updated every time a user joins the event or retires from it. This process keeps going on until the day that the *organizer* has selected as final date. In our example, it



Fig. 2. How to create an activity in *HappyMovie* and how events look like in *HappyMovie*

initially provides a recommendation for users U_1 , U_2 and U_3 when they first join the event. Later, when U_3 retires from the event a new recommendation is made for users U_1 and U_2 . Finally when U_4 , who was invited by U_2 , joins the event another new set of 3 movies appears for users U_1 , U_2 and U_4 . When the expiration date is reached users can see the final movie list. In that moment they are allowed to vote each of the movies individually. This process lets us decide which movie they are finally watching and, more important, it provides the required feedback to evaluate the quality of our recommendation.

2.2 Happy Movie’s Modular Architecture

The architecture of *HappyMovie* is represented in Figure 3. We can see that the application is divided in seven different modules: TKI Metaphor, Facebook Profile Analysis, Satisfaction Data Base, Web Test, Web Crawling, Content Based Estimation and *HappyMovie*’s group recommender. Next sections explain what are the basis of each of these modules.

- **TKI Metaphor:** This module fulfils the task of obtaining a value that represents the personality of each user. To do so, each user must answer to a personality test that measures people’s behavior in conflict situations. In our previous works [1,2,3] we used the TKI personality test [12], that consists on 30 questions where the user has to decide how she will react in the exposed situations. As we have prove in our previous works this is a tedious test to answer but its results are very reliable. To make the application more easy going we have introduced a movie metaphor as an alternative method to obtain the users personality in conflict situations. This interactive metaphor consists on displaying two movies characters with opposite personalities for five

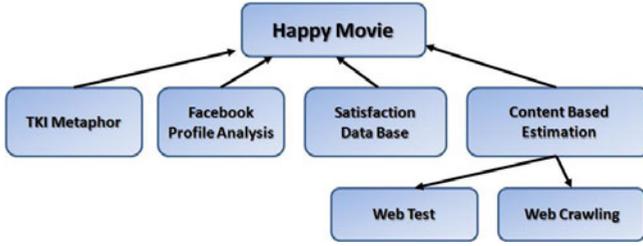


Fig. 3. Facebook application architecture

different modes of responding to conflict situations (this personality modes are the ones presented in the TKI experiment). One character represents the essential characteristics of the category, while the other one represents all the opposite ones. What the user has to do is to choose with whom of each pair of characters she feels more identified by simple moving a graded arrow, as seen in Figure 1 (left image).

- **Facebook Profile Analysis:** Once *Happy Movie* is running, the trust module must perform its estimation every time a user joins the event. When this happens the trust module explores the users who are currently in the list of attending users and calculates the trust of each of them with the user who has recently joined the event. To do so, the profiles of the two users will be analysed, comparing different social factors. The Trust Module is the module that has more benefits due to embedding the application in a social network. Previously, with a standalone application, the task of obtaining the data required to compute the trust between users was very tedious. Now, we are able to calculate the trust between users extracting the specific information from each of their own profiles in the social network. Users in Facebook can post on their profiles a huge amount of personal information that can be analysed to compute the trust to other users: distance in the social network, number of shared comments, likes and interests, personal information, pictures, games, duration of the friendship, ... We analyse 10 different trust factors comparing the information stored in their Facebook profiles. Next, these factors are combined using a weighted average. A detailed explanation of the trust factors obtained from Facebook and the combination process is provided in [3].
- **Satisfaction Data Base:** In this module we store all the recommendations that have been made for every user and every group. This specific feature of our application is fully detailed in Section 3.1. This module manages the following information: Each group to whom each user has participated on, each movie that each group has watched and the satisfaction of each user with each of her groups.
- **Web Test:** It consists on a test of the individual preferences of each user. Each time that the user uses the application she can modify her preferences or evaluate more, however it is only compulsory the first time she uses the application. In our specific domain, movies, the user will see the title of the

movie and the movie poster. Our users are provided with 50 movies from which they have to rate (in a rank of 0-5) at least 20, this is shown in Figure 1 (right image). These preferences are stored as the individual case base of each user.

- **Web Crawling:** This module searches the web and finds the movie listing of the city that the *organizer* has selected. Once it has that information it searches the complete file of each of the movies in the movie listing. Later, it analyses the file and extracts all the data required to define the movie. Each specific data is a field that the individual recommender contrasts. For example, in our particular case of study, these fields are main actors, director, year, etc. The recovered items, with all their specific information, are sent to the individual recommender module and to the group recommender module as they are the products to be recommended.
- **Content Based Estimation:** This is the individual recommender module, it is built using the jCOLIBRI framework extension to build recommender systems [13] and follows a content based approach [14] that uses descriptions of the products to be recommended and returns the collection of products that are more similar to the aimed product. In these particular case of study, *HappyMovie*, it returns the best three movies a user should watch individually. As it is a content based recommender system it has stored a case base, and the recommender compares all the considered items to be recommended with this case base. This case base is different for each user and has the information retrieved from the Web Test module.

3 Group Recommendation Methods

We have developed a group recommendation method which is based on the typical preference aggregation approaches. These approaches [5,15] aggregate the users individual predicted ratings $pred(u, i)$ to obtain an estimation for the group $\{gpred(G, i) | u \in G\}$. Then the item with the highest group predicted scoring is proposed.

$$gpred(G, i) = \bigsqcup_{\forall u \in G} pred(u, i) \tag{1}$$

Here G is a group of users, which user u belongs to. This function provides an aggregated value that predicts the group preference for a given item i . By using this estimation, *our group recommender proposes the set of k items with the highest group predicted scoring.*

In our proposal, we modify the individual ratings with the personality, trust and satisfaction factors. This way, we modify the impact of the individual preferences as shown in Equation 2.

$$gpred(G, i) = \bigsqcup_{\forall u \in G} pred'(u, i)$$

$$pred'(u, i) = \bigsqcup_{\forall v \in G} f(pred(u, i), p_u, t_{u,v}, s_u) \tag{2}$$

where $gpred(G, i)$ is the group rating prediction for a given item i , $pred(u, i)$ is the original individual prediction for user u and item i , p_u is the personality value for user u , $t_{u,v}$ is the trust value between users u and v , and s_u is the satisfaction of user u within the group.

A wide set of aggregation functions has been devised for combining individual preferences [16], being the average, least misery and most pleasure strategies the most commonly used:

- **Average Satisfaction:** Refers to the common arithmetic mean, which is a method to derive the central tendency of a sample space. It computes the average of the predicted ratings of each member of the group. The function that represents this strategy is:

$$gpred(G, i) = \frac{1}{|G|} \sum_{u \in G} pred'(u, i) \quad (3)$$

Where $pred'(u, i)$ is the predicted rating for each user u , and every item i . $gpred(G, i)$ is the final rating of item i for the group.

- **Least Misery:** This strategy considers that a group is as happy as its least happy member. The final list of ratings is the minimum of each of the individual ratings. A disadvantage can be that if the majority really likes one item, but one person does not, then it will never be chosen.

$$gpred(G, i) = \min_{u \in G} pred'(u, i) \quad (4)$$

- **Most Pleasure Strategy:** It is the opposite of the previous strategy, Least Misery, it chooses the highest rating for each item to form the final list of predicted ratings.

$$gpred(G, i) = \max_{u \in G} pred'(u, i) \quad (5)$$

Once we have introduced the typical aggregation approaches we can explain the estimation functions. We use three different methods to compute $pred'_{i,u}$, that as we have explained before, is a modification of the predicted rating for a user according to the personality, trust and satisfaction factors. The main ideas of these approaches are explained below:

- **Personality-based method (*pbm*):** Takes into account the differences in the personalities between pairs of individuals in the group. It is based on the modified average satisfaction employed in our previous work [1]. This strategy reflects that assertive characters will have more influence in the aggregated scoring than the cooperative characters. Our approach uses the type of personality to weight the influence of the estimated ratings during the recommendation process.
- **Delegation-based method (*dbm*):** The idea behind this method is that users create their opinion based on the opinions of their friends. It tries to simulate the following behaviour: when we are deciding which item to choose within a group of users we ask the people who we trust. Moreover, we also

take into account their personality to give a certain importance to their opinions (for example, because we know that a selfish person may get angry if we do not choose her preferred item).

- **Influence-based method (*ibm*):** Simulates the influence that each friend has in a given person. It supposes that the user may modify her preference for an item depending on the preferences of her friends to the same item. For example, if our rating for an item is 3 and our friend has a 5 rating for the same item, we could think on modifying our rating to 4. Depending on the trust with this friend, we decide the level of variation for our rating (i.e. 3.5 if the trust is low, and 4.5 if trust is high). Furthermore, the variation of our rating also depends on our personality. If we have a strong personality we will not be willing to change our rating, but if we have a weak personality we could be easily influenced by other users.

Once the estimation methods are outlined, next section details how to include the memory of users satisfaction in the recommendation process.

3.1 Including User Satisfaction in the Recommendation Process

Our approach for including fairness is based on the satisfaction of the users that conform a group. We propose a modification of the previous methods including a satisfaction parameter that measures the degree of happiness of every user regarding past recommendations. Our goal is to maximize the satisfaction among the group by promoting the items preferred by most dissatisfied users. To achieve this goal we need to keep track of past recommendations and the evolution of the satisfaction of each user. In *HappyMovie* this task is delegated to the Satisfaction Data Base module (see Figure 3).

Having recommendations with memory means that we are able to create a system that remembers all the previous recommendations for a given group. It is a Case-Based reasoning approach to the recommendation process, and a necessary step when providing a whole set of fair recommendations. This way, if one member accepts a proposal that she was not interested in, next time she will have some kind of priority in the recommendation process. This means that her opinion will have a higher weight next time. These weights will also be influenced by the different personalities of each group member. For example, someone with a strong personality that has been negatively affected would be immediately compensated next time; however someone with mild personality would not have problems giving in several times.

The satisfaction value s_u is the level of satisfaction of a user u . A user who is extremely happy with the recommendations will have this satisfaction measure value close to 1. However, the more upset with the recommendations she is, the more that this value will decrease, reaching down to 0 in the worst case. An important and interesting issue of this approach is the time scope of the memory of user's satisfaction. We can update the s_u value to reflect the satisfaction according to the last immediate group recommendation or to take into account previous ones. Therefore, the satisfaction value for an execution t of the recommender may depend on the satisfaction of the user with the items recommended

in t but it also depends on her satisfaction in the previous recommendations $t - 1, t - 2, \dots$. Therefore we manage two satisfaction values:

- Instant satisfaction ($i s_u$): reflects the immediate satisfaction of the user with the last recommendation. This is, her conformance with the last item recommended to the group. Its value can be obtained by estimating the preference of the user for the item selected to the group among all the items available.
- Global satisfaction: (s_u): measures the average satisfaction of the user among time. It is updated every time a recommendation is made:

$$s_u(t) = (1 - \delta) \cdot i s_u(t) + \delta \cdot s_u(t - 1) \quad (6)$$

In this equation we use the $\delta \in [0..1]$ parameter to adjust the impact of the previous satisfaction when updating that value. Somehow, this parameter measures the degree of forgetfulness about past (in)satisfaction. For example, some people could easily remember that they were not taken into account for the last recommendation when facing a new decision making process to select a similar item. On the other hand, other users won't ever take it into account. The measurement of this value belongs to the domain of the social sciences and is out of the scope of this paper. We have estimated it experimentally as exposed in next section.

It is important to note that the instant satisfaction value is also weighted depending on the personality of the user to reflect the importance of satisfying that concrete user.

In next Section we explain the details of our experimental evaluation to measure the impact of memory in the recommendation process of *HappyMovie*.

4 Case Study: Experimenting with Memory

Our goal is to estimate what is the best recommendation strategy for long term recommendations in *HappyMovie*. This strategy will be updated once we have real user data available. However, initially we must estimate this strategy to provide recommendations to our users. Therefore we have designed an experiment with synthetically generated data about users and movies. We must note that the validity of experimenting with this synthetically generated users has been already proven in our previous studies [3]. We simulate user preferences and personality to simulate different scenarios where several groups of users choose a movie for going to the cinema. In our previous work [4] we have evaluated the estimation strategies –personality ($p b m$), delegation ($d p m$) and influence($i b m$)– without taking into account the memory of past recommendations. These experiments were performed with data from real volunteers that simulated going to the cinema together. Results shown that $d b m$ and $i b m$ provide better results than $p b m$ when including the trust factor (t_u). However for our simulation it is impossible to generate synthetically that value and therefore, we have focused on the $p b m$ to evaluate which is the best aggregation function: average satisfaction, least misery or most pleasure.

4.1 Experimental Set-Up

The experiment configuration runs as follows. We have simulated 1000 groups of individuals going to the movies together 15 times. Each group consists of 10 individuals. Although the composition of the group does not change, having such a large number of groups let's us include in the simulation any kind of variation in their composition. Movies are described by means of a vector that represents the degree of conformance with several genres (terror, action, romance). These genres were obtained from the MovieLens database [17]. Correspondingly, cinema preferences of each individual are represented in the same way. Movies and individual preferences are generated randomly and compared using the cosine distance to obtain what would be the real rating of an individual for a given movie. This real rating (referred as $rr(u, m)$ with range $[0..10]$) will be later used to evaluate the performance of our recommender.

Our recommendation method is based on an individual recommender that estimates the rating $pred(u, m)$ given by a user to a movie. This recommender has been implemented using the jCOLIBRI framework [13] and follows the collaborative filtering approach described in [18] based on the Pearson Correlation. This method requires a population of previous individuals that have rated several items (movies). These users and their ratings are also generated synthetically.

Finally, the personality of each user is assigned according to the probability distribution inferred from the 50 volunteers that took part in our first experiments. It is important to note that we could also apply the distribution used by the original TKI test¹.

On each round of the simulation (there are 15 rounds per group) we generate a random movie listing composed of 10 movies. Our group recommender predicts what is the best movie for the group $gpred(G, m)$. Then, the proposed movie is compared with the real preferences of each individual to compute their instant satisfaction and the global satisfaction of the group. To obtain the instant satisfaction we order the movie listing for each individual according to the real rating that she would assign to each movie. Next we compare what is the position of the movie selected for the group in that list. Instant satisfaction will be higher if $gpred(G, m)$ is in the first positions and lower it is at the end of the list. Instead of using directly the position of the movie recommended for the group in the individual ordered list, we slightly modify that position according to the personality of the individual. A user with a strong personality won't be happy if the movie is in the second or third position of her preference list because she will expect to see her first favourite movie. On the other hand, an individual with mild personality won't mind to watch a movie in the middle of her preferences list. We refer to this value as the *dislike* factor. And it is linearly weighted to compute the instant satisfaction:

$$is_u = (size(ml) - dislike(u, m)) / size(ml) \quad (7)$$

$$dislike(u, m) = index(m, ml) + p_u \quad (8)$$

¹ TKI personality distribution is obtained from a population of 8000 individuals from U.S.A.

where ml is the movie listing proposed to the users, and $size(ml)$ returns its size. $m \in ml$ is the movie recommended by our system for the group. The position of a movie in the movie listing once ordered according to the user preferences is obtained by means of the $index(m, ml)$ function. Finally, p_u is the personality of the user with range $[-1, 1]$.

Once is_u is obtained, the global satisfaction s_u is obtained. We have configured a δ value of 0.5 to represent balanced impact of previous satisfaction.

4.2 Results

We have run the experiment with the three aggregation functions: average satisfaction (AS), least misery (LM) and most pleasure (MP). These aggregation functions combine the individual prediction for each movie. This prediction is obtained by means of the Personality-based method. It is computed as follows: (1) the individual estimation of the rating given to the movie is returned by the collaborative individual recommender. (2) This rating is weighted according to the personality of the user p_u . (3) Resulting rating is again modified according to the user satisfaction s_u . Step (3) tries to promote those movies that have a high estimated rating for an unsatisfied user. On the other hand it decreases the final rating of movies with low estimation (to avoid their selection for the group). If a user is satisfied, ratings are slightly modified. Analogously, step (2) takes into account the personality of the user to promote the movies preferred by users with strong personalities.

In our evaluation we have studied the effect of the previous modifications. We refer to BASE when we only apply step (1) to obtain the individual prediction (note that it is the standard aggregation function and the baseline of our metric). The measures including only steps (1) and (2) are referred as PERSONALITY as they only take into account the personality factor. Finally, the complete method including steps (1), (2) and (3) is named as (PERSONALITY_MEMORY) because it includes both the personality factor and the satisfaction memory.

To measure the performance of the group recommender we use the average accumulative satisfaction of the group. The accumulative satisfaction is the sum of the individual satisfaction s_u of a user after n cinema events. This way, a user that had a high satisfaction in several events will finish with a high accumulative satisfaction and a user that was not taken into account will have a low value. To reflect the satisfaction of the group we compute the mean of the accumulative satisfaction of every user in that group.

However, our goal is to maximize the mean satisfaction but minimizing the standard deviation. The mean represents the global satisfaction of the group and reflects the goodness of the items recommended by our system. On the other hand, the standard deviation reflects the differences in the satisfaction levels within the group. It is a measure of the social fairness. As conclusion, our evaluation function is the mean value minus the standard deviation of the accumulative satisfaction of the group $(\bar{x} - \sigma)$, where a higher is better.

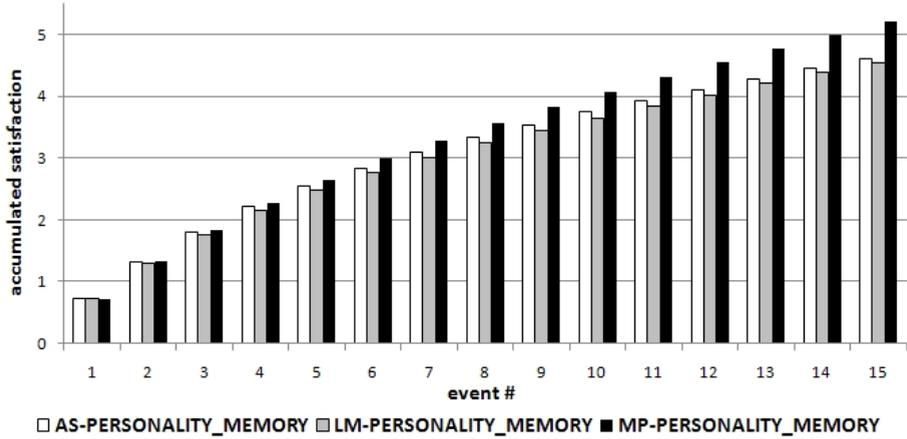


Fig. 4. Evaluation results

Results of this evaluation function are shown in Figure 4. For clarity reasons BASE and PERSONALITY methods are not shown. All of them obtain worse results than the PERSONALITY_MEMORY approach. Using just personality is slightly better than the BASE methods. As we could expect, the inclusion of personality and memory provides the best performance, being the most pleasure (MP) aggregation function the optimal approach. It is followed by average satisfaction (AS) and finally we find least misery (LM). This behaviour can be explained thinking about the nature of these aggregation functions and the bias that we promote when including the memory of user satisfaction. The main consequence of including the memory of user satisfaction is that we minimize the standard deviation within the group (i.e. maximize the fairness). By definition, LM gives not very good results in average but maximizes the fairness. However, AS and MP provide respectively good and very good recommendations in average but don't care about the fairness. With the inclusion of the satisfaction in the estimation functions we remove this drawback found in AS and MP. Therefore, they provide high values in average and minimize the standard deviation. Concretely, MP returns the item with the higher rating for the user, even if that item is a very worse recommendation for other users. This fact maximizes the mean satisfaction, and our bias ensures that others' satisfaction does not decrease too much. This is the reason it obtains the highest performance.

When analysing the mean and the standard deviation separately, we have not found very significant differences in the mean satisfaction. However the standard deviation is highly influenced by the inclusion of the memory. We have also measured the impact of modifying the size of the group, the movie listing length and the number of events, not finding any correlation between these variables and the system performance.

5 Conclusions

In this paper we have introduced our Facebook application Happy Movie. It is a group recommender for the movies domain that takes advantage of the social variables available in social networks that can be exploited to improve the performance of the system. We propose the inclusion of the following social factors: personality of every group member, trust between users, and a memory of users satisfaction to promote fairness. In this paper we have focused in this last factor –memory of users satisfaction– as we propose a CBR approach to modify the items presented to the group depending on the evolution of this satisfaction.

Our approach can be applied with several aggregation functions –that provide global recommendation for the group– and different estimation measures that predict the rating a user would assign to a given item. We have run an experiment with synthetic data to obtain the best approach for the *HappyMovie* application. Results show that optimal performance is obtained by means of the most pleasure aggregation function together with the inclusion of personality and memory in the estimation. Our future work consists on confirming these results with real users that could use our application to organize their cinema events.

Acknowledgments. Supported by Spanish Ministry of Science & Education (TIN2009-13692-C03-03) and Madrid Education Council and UCM (Group 910494).

References

1. Recio-García, J.A., Jimenez-Diaz, G., Sánchez-Ruiz, A.A., Díaz-Agudo, B.: Personality aware recommendations to groups. In: Bergman, L.D., Tuzhilin, A., Burke, R.D., Felfernig, A., Schmidt-Thieme, L. (eds.) *Procs. of the 2009 ACM Conference on Recommender Systems*, pp. 325–328. ACM, New York (2009)
2. Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B.: Social based recommendations to groups. In: *Procs. of the 14th UK Workshop on Case-Based Reasoning*, pp. 46–57. CMS Press, University of Greenwich (2009)
3. Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B.: Personality and social trust in group recommendations. In: *Procs. of the 22th International Conference on Tools with Artificial Intelligence, ICTAI 2010* (to appear, 2010)
4. Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B., Jiménez-Díaz, G.: Social factors in group recommender systems. In: *ACM-TIST, TIST-2011-01-0013* (to be published, 2011)
5. Masthoff, J., Gatt, A.: In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. *User Modeling and User-Adapted Interaction* 16, 281–319 (2006)
6. Barsade, S.G.: The ripple effect: Emotional contagion and its influence on group behavior. *Administrative Science Quarterly* 47, 644–675 (2002)
7. Hatfield, E., Cacioppo, J., Rapson, R.: *Emotional Contagion. Studies in Emotion and Social Interaction*. Cambridge University Press, Cambridge (1994)

8. McCarthy, J.F., Anagnost, T.D.: MusicFX: An arbiter of group preferences for computer supported collaborative workouts. In: CSCW 1998: Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work, pp. 363–372. ACM, New York (1998)
9. Crossen, A., Budzik, J., Hammond, K.J.: Flytrap: intelligent group music recommendation. In: IUI, pp. 184–185 (2002)
10. Baccigalupo, C., Plaza, E.: A case-based song scheduler for group customised radio. In: Weber, R., Richter, M.M. (eds.) ICCBR 2007. LNCS (LNAI), vol. 4626, pp. 433–448. Springer, Heidelberg (2007)
11. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variants, and system approaches. *Artificial Intelligence Communications* 7, 39–59 (1994)
12. Thomas, K., Kilmann, R.: Thomas-Kilmann Conflict Mode Instrument, Tuxedo, N.Y. (1974)
13. Díaz-Agudo, B., González-Calero, P.A., Recio-García, J.A., Sánchez-Ruiz-Granados, A.A.: Building cbr systems with jcolibri. *Sci. Comput. Program.* 69, 68–75 (2007)
14. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) *Adaptive Web 2007*. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007)
15. O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J.: Polylens: a recommender system for groups of users. In: ECSCW 2001: Proceedings of the Seventh Conference on European Conference on Computer Supported Cooperative Work, pp. 199–218. Kluwer Academic Publishers, Norwell (2001)
16. Masthoff, J.: Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Modeling and User-Adapted Interaction* 14, 37–85 (2004)
17. Bobadilla, J., Serradilla, F., Hernando, A.: Collaborative filtering adapted to recommender systems of e-learning. *Knowl.-Based Syst.* 22, 261–265 (2009)
18. Kelleher, J., Bridge, D.G.: An accurate and scalable collaborative recommender. *Artif. Intell. Rev.* 21, 193–213 (2004)