**Applied Intelligence manuscript No.**
(will be inserted by the editor)

# An Architecture and Functional Description to Integrate Social Behaviour Knowledge Into Group Recommender Systems

**Lara Quijano-Sanchez · Juan A. Recio-Garcia · Belen Diaz-Agudo**

the date of receipt and acceptance should be inserted later

**Abstract** In this paper we consider the research challenges of generating a set of recommendations that will satisfy a group of users, with potentially competing interests. We review different ways of combining the preferences of different users and propose an approach that takes into account the social behaviour within a group. Our method, named *delegation-based prediction method*, includes an analysis of the group characteristics, such as size, structure, *personality* of its members in conflict situations, and *trust* between group members. A key element in this paper is the use of social information available in the Web to make enhanced recommendations to groups. We propose a generic architecture named ARISE (Architecture for Recommendations Including Social Elements) and describe, as a case study, our Facebook application *HappyMovie*: a group recommender system that is designed to provide assistance to a group of friends that might be selecting which movie to watch on a cinema outing. We evaluate the performance (compared with the real group decision) of different recommenders that use increasing levels of social behaviour knowledge.

**Keywords** Group Recommender systems, Social Networks, Personality, Trust, Generic Architecture

## 1 Introduction

It is becoming common to employ recommendation technologies to aid users in the task of finding interesting items in the Web [37]. There is a wide range of products such as books, music, games, trips, etc. that are difficult to discover in the Web due to the overwhelming amount of information available. Recommender systems [24] enable users to find items and provide a richer and more interactive user experience than classical interfaces based on catalogues of products.

Initially, existing recommenders were focused on individual users [16,31]. Nowadays, however the rise of the collaborative Web (a.k.a. Web 2.0) has encouraged the

Facultad de Informatica. Universidad Complutense de Madrid.
C/ Prof. Jose Garcia Santesmases s/n. 28040. Madrid, Spain.
E-mail: {lara.quijano,jareciog,belend}@fdi.ucm.es

development of activity-planning through social networks, like watching a movie, going to a restaurant, listening to a radio station or traveling with friends. A clear example are events organized through social networks like Facebook. Here, recommender systems can play a significant role, since agreement on a common item by several users is not a simple task. To address this issue, the number of recommender systems that deal with the challenge of making recommendations for groups of people has increased [38,34]. Group recommendation, however, is not a mere aggregation of individual preferences. Humans are social individuals and, therefore, social behaviour has a great impact on their group decision-making processes. Our proposal takes into account this fact and assumes that the general satisfaction of the group does not always mean aggregating its members' preferences. It is clear that groups have an influence on individuals when coming to a decision. This is commonly referred to as *emotional contagion*: the effect of individuals' affective state on others in the group [33,6,21]. This contagion is usually proportional to the *tie strength* or *trust* between individuals as closer friends have a higher influence [50,19,39]. However, the influence of the group also depends on the individual's degree of *conformity* [33]. It has been proved that humans adjust their opinions to conform with those of a group when the majority of the group expresses a different opinion. The degree of conformity is counteracted by the individual's behaviour when facing a conflict situation. Here, *personality* influences the acceptance of others' proposals [44].

Previous research on group recommendation consider the preferences of every member in the group with the same degree of importance and try to satisfy the preferences of every individual. However, all these social elements (emotional contagion, trust, personality, . . . ) should be included in the recommendation model to fully represent the group behaviour when choosing a shared item. Although it seems natural to model this social knowledge, a major limitation appears: social factors are very difficult to estimate. Up to now, it was impossible to obtain these factors without annoying users with several questionnaires. But nowadays the collaborative Web provides a tool that can be used to lighten this problem: social networks. Social networks let users interact and develop their social relationships in a computer-based environment. Indeed, several works have pointed out that social elements can be inferred from them [20,9]. For example, we can estimate a tie between users by measuring the number of messages exchanged or the number of friends in common.

The first contribution of this paper is the compilation of our ideas in an organized generic architecture named ARISE (Architecture for Recommendations Including Social Elements) that can be instantiated into group recommender systems that take into account social behaviour knowledge. In the functional description of our architecture we will detail how social knowledge provided by the modules inside ARISE is combined to obtain a recommendation that integrates the individual preferences and social features of the group.

To do so, individual preferences are modified according to the social environment of the user. This idea is reflected in our novel technique to estimate an individual's preference for a given item based on social factors. A preliminary version of this technique was introduced in [42]. The research presented in this paper shows a more mature work, where we have refined, tested and justified the ideas and decisions made in [42]. We have named this new approach the *delegation-based prediction* (*dbp*) method. As the name suggests, the idea behind this method is

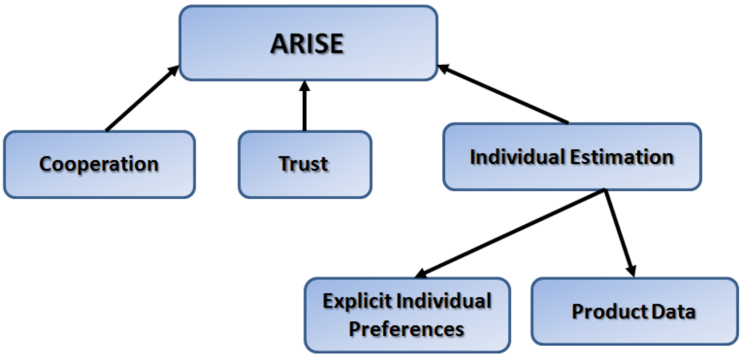


**Fig. 1** Overview of ARISE

that users create their preferences based on others' opinions. We consider this new perspective of our past work as the second contribution of the paper.

In the next step, these individual predictions are combined to generate an aggregated preference for the group. Masthoff [32] presents a compilation of the most important preference aggregation techniques pointing out that the selection of a proper aggregation strategy is a key element in the success of the generated recommendation for the group. Therefore, our third contribution is the adaptation of these techniques to our *delegation-based* method, plus a comparative analysis of their performance. This study indicates which is the best aggregation strategy depending on the characteristics and nature of the group.

To perform this evaluation we have instantiated our generic architecture into a real application called *HappyMovie*, that conforms the last contribution of this paper. It is a Facebook system for the movie recommendation domain. Although we have chosen this domain as a case study, we discuss how the architecture and group recommendation approaches presented in this paper could be applied to any other domain.

The discussion about this architecture and its instantiation is presented first in Section 2 (ARISE) and Section 4 (*HappyMovie*). Next, Section 3 includes the functional description of ARISE and introduces the delegation-based prediction method. The experimental evaluation and comparative analysis of this method together with the aggregation strategies is presented in Section 5. Finally, Section 6 introduces related work on group recommender systems and Section 7 concludes the paper.

## 2 Generic architecture for group recommenders using social elements

ARISE[1] is a theoretical organization of the modules required to build social group recommenders. The architecture of ARISE is represented in Figure 1. We can see that it is divided into six different modules: *cooperation*, *trust*, *individual prediction*, *explicit individual preferences*, and *product data*. This architecture allows us to simulate, in a realistic way, the social behaviour followed by groups of people when arguing on a joint activity.

[1] Architecture for Recommenders Including Social Elements

The *personality* factor lets us model the behaviour of each member in a conflict situation such as the ability to agree on a common group activity. During this decision-making process each member must give up some preferences to reach a consensus. This preference variation is directly influenced by the confidence or social *trust* in other members of the group.

A basic building block of our group recommender is the individual estimation module that predicts the preferences for a given user. It requires an *explicit profile of the individual's preferences* and a *product data* set to be recommended. As we will see in Section 3.2, our *delegation-based prediction method* biases these individual estimations according to the personality and trust factors.

Finally, the information provided by each module is combined by the ARISE's aggregation techniques to obtain a recommendation for the group. These combination strategies are explained in Section 3 whereas the architecture modules are explained in the following subsections.

## 2.1 Cooperation Module

It is a fact that when we face a situation in which the concerns of people appear to be incompatible, *conflict situations* arise. Different people have different expectations and behaviour in conflict situations, and therefore they should be taken into account. When we started our research to improve the group recommendation process, we decided to study the different behaviours that people have in conflict situations according to their personality [44,41,42].

This module fulfils the task of obtaining a value that represents the personality of each user. This personality value, $p_u$, fits within a range of (0,1], 0 being the reflection of a very cooperative person and 1 the reflection of a very selfish one. In the ARISE architecture it is described as a high-level module that can be implemented in different ways depending on the resources available and the domain of the recommender application.

## 2.2 Trust Module

Current research has pointed out that people tend to rely more on recommendations from people they trust (friends) than on recommendations based on anonymous ratings [47]. This social element is even more important when we are performing a group recommendation where users have to choose an item for the whole group. Note that *trust* is also related to *tie strength* and previous works have reported that both are conceptually different but there is a correlation between them [28].

This module fulfils the task of obtaining the trust values, $t_{u,v}$, between every user $u$ and $v$ that belong to the group that is being recommended. Note that $t_{u,v}$ $\epsilon$ (0,1], 0 being the reflection of a person not to be trusted and 1 the reflection of a highly trusted one.

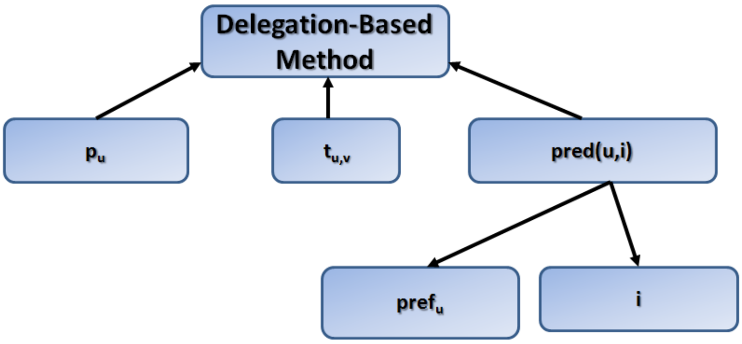


**Fig. 2** Functional Description of ARISE

### 2.3 Individual estimation

Our recommendation approach predicts the rating that each user would assign to every item in the catalogue and then these estimated ratings are aggregated to obtain a global prediction for the group. Therefore, a basic building block of the architecture is the module in charge of computing individual predictions. We will denote the individual predicted rating as: $pred(u, i)$, $u$ being a user and $i$ an item from the catalogue. There are several options for obtaining these predictions that have been broadly studied in the recommendation research. In a general way there are two different approaches [45]. *Collaborative recommenders* use ratings already assigned by other users to several products. Users are selected according to their similarity with the target individual (by comparing the ratings given to the products). Most similar users are used as predictors and their ratings are combined to estimate the rating that the target user would assign to a new product. On the other hand, the *Content-based approach* compares each item to be proposed with items already rated by the target user. Then the ratings of the most similarly rated items are combined to provide a prediction.

Regardless of the approach chosen to implement this generic module of the ARISE architecture, there are two components (or sub-modules) that are always required by the individual recommender: explicit individual preferences and the product data set. Explicit individual preferences span any kind of information about the user that is required to predict the rating for a new item. Commonly, it just consists of the ratings given to some products in the catalogue. These ratings will be used later by the collaborative or content-based approach to predict new ratings. The product data module provides the information about the items in the catalogue that should be recommended to the group.

The next section presents the functional description of our approach.

## 3 Functional Description of ARISE

In this section we explain the process of combining the social knowledge obtained from each of the ARISE's modules in order to provide a recommendation for the group.

| $\mathbf{dbp(u,i)}$ | Item A | Item B | Item C | Item D | Item E | Item F |
|---|---|---|---|---|---|---|
| $u_1$ | 5 | 2 | 1.5 | 3.5 | 5 | 4.5 |
| $u_2$ | 0.5 | 4.5 | 4 | 4.5 | 3.5 | 4.5 |
| $u_3$ | 5 | 2.5 | 1 | 3.5 | 4.5 | 4 |

**Table 1** Example of a possible estimation of the ratings given by every user for different items. (Note that this table shows the values once they have been modified by our method to reflect personality and trust).

Although we have already explained these modules, we summarize their output values to introduce some notation. These values and its corresponding dataflow are displayed in Figure 2. The *cooperation* module obtains a personality factor $p_u$; the *trust* module returns the trust factor $t_{u,v}$; the *individual prediction* module obtains $pred(u,i)$ that is the result of applying a content-based predictor that compares the user's preferences $pref_u$, given by the *explicit individual preferences* module, to the *product data* module, that stores every item $i$ in the catalogue.

Our group recommendation method is based on preference aggregation approaches. These approaches [33, 38] aggregate the individual ratings, predicted for every user $u$ given an item $i$ -denoted as $pred(u,i)$-, to obtain a prediction for the group:

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

Here $G$ is a group of users, which user $u$ belongs to, and $pred(u,i)$ is the individual prediction for user $u$ and item $i$ returned by the individual estimation module. There are several aggregation functions -represented with the $\sqcup$ symbol- that can be chosen to obtain the group prediction. These functions provide an aggregated value that predicts the group preference for a given item $i$. Then, our group recommender proposes the k items with the highest estimated group scoring.

As we previously explained, individual predictions are biased by our *delegation-based prediction* method that takes into account the personality and trust factors. This way, recommendations are computed as shown in Equation 2.

$$gpred(G,i) = \bigsqcup_{\forall u \in G} dbp(u,i) \tag{2}$$

We will explain next the aggregation functions and later the *delegation-based* prediction method.

### 3.1 Aggregation Functions

A wide set of aggregation functions has been devised to combine individual preferences [32]. Choosing the aggregation function that performs best is a key element in providing good recommendations. Here we explain the functions that we have studied for our social prediction method, *dbp*, which will be elaborated on in Section 3.2. We explain how to calculate group ratings with each of these methods through an example. Table 1 contains an example of predicted individual ratings returned by the *dbp* method, whereas Tables 2 to 8 show how these individual predictions are modified and/or combined in order to get the final group recommendation.

| **Avg. Sat.** | Item A | Item B | Item C | Item D | Item E | Item F |
|---|---|---|---|---|---|---|
| Group Prediction | 10.5/3 | 9/3 | 6.5/3 | 11.5/3 | 13/3 | 13/3 |

**Table 2** Example of aggregation with Average Satisfaction (from ratings in Table 1). Predicted group preference: E,F $\succ$ D $\succ$ A $\succ$ B $\succ$ C

| **Borda Count** | Item A | Item B | Item C | Item D | Item E | Item F |
|---|---|---|---|---|---|---|
| $u_1$ | 4.5 | 1 | 0 | 2 | 4.5 | 3 |
| $u_2$ | 0 | 4 | 2 | 4 | 1 | 4 |
| $u_3$ | 5 | 1 | 0 | 2 | 4 | 3 |
| Group Prediction | 9.5 | 6 | 2 | 8 | 9.5 | 10 |

**Table 3** Example of aggregation with Borda Count (from ratings in Table 1). Predicted group preference: F $\succ$ A,E $\succ$ D $\succ$ B $\succ$ C

| **Copeland Rule** | Item A | Item B | Item C | Item D | Item E | Item F |
|---|---|---|---|---|---|---|
| Item A | 0 | -1 | -1 | -1 | 0 | -1 |
| Item B | +1 | 0 | -1 | +1 | +1 | +1 |
| Item C | +1 | +1 | 0 | +1 | +1 | +1 |
| Item D | +1 | -1 | -1 | 0 | +1 | +1 |
| Item E | 0 | -1 | -1 | -1 | 0 | -1 |
| Item F | +1 | -1 | -1 | -1 | +1 | 0 |
| Group Prediction | +4 | -3 | -5 | -1 | +4 | +1 |

**Table 4** Example of aggregation with Copeland Rule (from ratings in Table 1). Predicted group preference: A,E $\succ$ F $\succ$ D $\succ$ B $\succ$ C

| **Approval Voting** | Item A | Item B | Item C | Item D | Item E | Item F |
|---|---|---|---|---|---|---|
| $u_1$ | 1 |  |  | 1 | 1 | 1 |
| $u_2$ |  | 1 | 1 | 1 | 1 | 1 |
| $u_3$ | 1 | 1 |  | 1 | 1 | 1 |
| Group Prediction | 2 | 2 | 1 | 3 | 3 | 3 |

**Table 5** Example of aggregation with Approval Voting ($\delta = 2.5$) (from ratings in Table 1). Predicted group preference: D,E,F $\succ$ A,B $\succ$ C

– **Average Satisfaction:** Refers to the common arithmetic mean, which is a method to derive the central tendency of a sample space [1]. It computes the average of the predicted ratings of each member of the group. The function that represents this strategy is:

$$gpred(G,i) = \frac{1}{|G|} \sum_{u \in G} dbp(u,i) \tag{3}$$

Where $dbp(u,i)$ is the socially modified predicted rating for each user $u$, and every item $i$. $gpred(G,i)$ is the final prediction of item $i$ for the group. An example of this strategy is shown in Table 2.

– **Borda Count:** The Borda count is a single-winner election method in which users rank candidates in order of preference [46]. The Borda count determines the winner of an election by giving each candidate a certain number of points corresponding to the position in which s/he is ranked by each voter. Once all votes have been counted the candidate with more points is the winner. Because it sometimes elects broadly acceptable candidates, rather than those preferred by the majority, the Borda count is often described as a consensus-based electoral system, rather than a majoritarian one. We can see how the Borda count measure works in our example in Table 3. For instance, $u_1$ has

| **Least Misery** | Item A | Item B | Item C | Item D | Item E | Item F |
|---|---|---|---|---|---|---|
| Group Prediction | 0.5 | 2 | 1 | 3.5 | 3.5 | 4.5 |

**Table 6** Example of aggregation with Least Misery (from ratings in Table 1). Predicted group preference: F $\succ$ E,D $\succ$ B $\succ$ C $\succ$ A

| **Most Pleasure** | Item A | Item B | Item C | Item D | Item E | Item F |
|---|---|---|---|---|---|---|
| Group Prediction | 5 | 4.5 | 4 | 4.5 | 5 | 4.5 |

**Table 7** Example of aggregation with Most Pleasure (from ratings in Table 1). Predicted group preference: A,E $\succ$ B,D,F $\succ$ C

| **Avg. w/out Misery** | Item A | Item B | Item C | Item D | Item E | Item F |
|---|---|---|---|---|---|---|
| Group Prediction | - | 9/3 | - | 11.5/3 | 13/3 | 13/3 |

**Table 8** Example of aggregation with Avg. Without Misery, $\delta = 2$ (from ratings in Table 1). Predicted group preference: E,F $\succ$ D $\succ$ B

the lowest rating for C, and hence, C is awarded 0 points. Next rating is for item B and it gets 1 point, and so on with the rest of its rankings. Finally, to obtain the group preference order, the points awarded to the individuals are added up.

$$\begin{aligned} gpred(G,i) &= \sum_{u \in G} bs(u,i) \\ bs(u,i) &= pos(\ i\ ,\ OL(u)\ ) \\ OL(u) &= \{i_1, , i_2, \ldots, i_n\} \\ &\quad \text{where } dpb(u,i_p) \leq dbp(u,i_{p+1}) \end{aligned} \tag{4}$$

Where bs(u,i) is the Borda score assigned to each item rated by user $u$. It is obtained as the position of the item $i$ in the ordered list $OL$. This list arranges the items according to the ranking estimated for user $u$. A problem arises when an individual has multiple alternatives with the same rating. In this case we have decided to distribute the points. So, for example, in $u_2$'s list B, D and F share the place and get (3+4+5)/3=4 points each. (Note that this modification is not included in the previous formula for the sake of readability).

- **Copeland Rule:** Alternatives are ordered by the number of pairwise victories, minus the number of pairwise defeats. It is a good procedure to overcome problems resulting from voting cycles [26]. In the example A beats B as both $u_1$ and $u_3$ prefer it, so the result in Table 4 shows a +1 for column A vs row B.

$$\begin{aligned} gpred(G,i) &= \sum_{j \in Catalog, j \neq i} cs(i,j) \\ cs(i,j) &= \begin{cases} +1 & \text{if } wins(i,j) > losses(i,j) \\ -1 & \text{if } wins(i,j) < losses(i,j) \\ 0 & \text{a.o.c.} \end{cases} \\ wins(i,j) &= |u \in U : dbp(u,i) > dbp(u,j)| \\ losses(i,j) &= |u \in U : dbp(u,i) < dbp(u,j)| \end{aligned} \tag{5}$$

- **Approval Voting:** This is a single-winner voting system used for elections. Each voter may vote for (approve of) as many of the candidates as they wish. The winner is the candidate that receives more votes [17]. In our example, we could assume that $u_1$, $u_2$ and $u_3$ vote for all alternatives with a rating above a certain threshold $\delta$, meaning that they vote for any alternative provided that it seems a little interesting for them. An example of this strategy is reflected in Table 5 with $\delta = 2.5$.

$$\begin{aligned} gpred(G,i) &= \sum_{u \in G} as(u,i) \\ as(u,i) &= \begin{cases} 1 & \text{if } dbp(u,i) \geq \delta \\ 0 & \text{a.o.c.} \end{cases} \end{aligned} \tag{6}$$

- **Least Misery:** This strategy follows the idea that, even if average satisfaction is high, a solution that leaves one or more members very dissatisfied is likely to be considered undesirable. 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 even if the majority really likes one item, if one person does not, then it will never be chosen [32]. An example of this is shown in Table 6 where $u_1$ and $u_3$ vote very highly for item A but its final rating is the lowest one, because $u_2$ does not like it.

$$gpred(G,i) = \min_{u \in G} dbp(u,i) \tag{7}$$

- **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 [32], as we can see in Table 7.

$$gpred(G,i) = \max_{u \in G} dbp(u,i) \tag{8}$$

- **Average Without Misery:** Assigns a preference to the average of the weights in the individual ratings. The difference here is that those items that have predicted ratings under a certain threshold will not be considered [32]. Table 8 shows an example of how the group ratings are calculated using a threshold of $\delta = 2$.

$$\begin{aligned} gpred(G,i) &= \frac{\sum_{u \in G} pred_{wm}(u,i)}{|u \in U : dbp(u,i) > \delta|} \\ pred_{wm}(u,i) &= \begin{cases} dbp(u,i) & \text{if } dbp(u,i) > \delta \\ 0 & \text{a.o.c.} \end{cases} \end{aligned} \tag{9}$$

Once we have described the aggregation functions that can be used to combine individual predictions, the following subsection details how these individual estimated ratings are modified with our social factors. We present, as core of ARISE, our *delegation-based* method, that improves group recommendations by means of personality and trust factors.

### 3.2 Modifying individual predictions with social elements

The idea adopted in our method is that everyone is influenced by their social context. Social media highly influences our decisions, relationships, and education. Several researchers study the impact of social media in our lives [12]. The social context, refers to the immediate physical and social setting in which people live. It includes the culture that the individual was educated or lives in, and the people and institutions with whom they interact. Circumstantial life events, influences, and surroundings can further change our behaviour [7]. Social elements, that in our social recommendation method are the personality and trust factors, define each person (our users involved in the recommendation processes) as a potentially influenced component of a social community or group determined by the environment, in most cases social networks, s/he belongs to. In our social method, we have simulated people's behaviour based on the idea that the relationship between individuals and their networks of people directly influence their lives [12]. This way, we use the trust factor to model the impact of the preferences of the people that belong to the close circle of the user in her/his social environment and that therefore might influence her/him. This proximity between users (users trust) is obtained by analyzing the information available on the social network. But, the influence of other group members not only depends on their proximity or trust in them, but also in the degree of personality or leadership of these influencers and in the degree in which the user might be influenced according to her/his personality. This degree of compliance or leadership is computed through the personality factor with the *assertiveness* and *cooperativeness* dimensions (as we will explain next in Section 4.1).

Hence, our recommendation approaches consist of evaluating the different behaviours that people have when participating in a decision-making process. To do so we use the personality and trust factors to modify the predictions made by the individual recommender. In that way not all the predictions are taken into account equally. We use a novel approach, which we have named *delegation-based* prediction method, to compute the new individual prediction, $dbp(i,u)$, used in Equation 2.

The idea behind this approach is that users create their opinions based on their friends' opinions. So basically, in each user's turn in $\forall u \in G, |G| = n$ in Equation 2, the user's opinion is not taken into account but in the other (n-1) turns that is when the user influences others. Instead of storing the information contained in a user's opinion just once, the method takes it into account every time another user of the same group states an opinion. We know that this idea is not at all intuitive. However, we performed several experiments with other simpler methods and they all provided worse recommendations than our *dbp* method. The delegation-based prediction method tries to simulate the following behaviour: when we are deciding which item to choose within a group of users we ask people whom we trust. This method follows a collaborative approach where a user's opinion is generated based on others' preferences. This way we apply the principles of emotional contagion. Moreover, we also take into account their personality in order to give certain importance to their opinions (for example, because we know that a selfish person may get angry if we do not choose her/his preferred item). The tie strength is also reflected in the formula by means of the trust between the users. The delegation-based prediction, $dbp(u,i)$, given a user $u$ and an item $i$ is computed in this way:

$$dbp(u,i) = \frac{1}{T} \sum_{v \neq u \in G} t_{u,v} [\, pred(v,i) + \theta_{v,i} \cdot (p_v - p_u) \,] \tag{10}$$

where

$$T = \sum_{v \neq u \in G} t_{u,v}$$

In this formula, we take into account the predicted preference $pred(v,i)$ of every friend $v$ for item $i$. This rating is increased or decreased depending on the differences of personality between both friends, $p_v - p_u$. This way if user $v$ has a strong personality s/he will have a higher impact on the prediction for user $u$. However, it is important to note that a user $v$ with a strong personality and a high preference for item $i$, $pred(v,i)$, would try to increase the opinion of user $u$ about that item. In the opposite case, a low preference for the item, user $v$ would try to decrease $u$'s opinion. This behaviour is modeled using the $\theta_{v,i}$ parameter as follows, lets say that $pred(v,i)$ is in a range of [a,b]:

$$\theta_{v,i} = \begin{cases} 5 & \text{if } pred(v,i) \geq \frac{b-a}{2} \\ -5 & \text{if } pred(v,i) < \frac{b-a}{2} \end{cases} \tag{11}$$

We have chosen those constant values (5 and -5) because the mean difference in the personality values is 0.2 and therefore the impact of the difference of personality in the formula will be $\sim \pm 1$. Finally, the prediction of user $v$ that has been modified according to the personalities is also weighted by the trust between both users $t_{u,v}$. Note that this formula is not normalized by the group size and uses the accumulated trust[2] (represented as $T$). We have chosen this option following the findings of [20] where a method for group recommendations using trust is proposed.

We will now explain the details of our case study *HappyMovie*[3]: a Facebook application for recommending movies to groups of users.

## 4 Case Study: HappyMovie

*HappyMovie* is a particular instantiation of our generic ARISE architecture for the movie recommendation domain. It serves as a use case and experimental environment where we can evaluate our architecture with real products. This way we can validate and improve our previous results obtained in simulated environments [44, 41,42].

This application has been developed for Facebook. With it we are able to offer group recommendations to people connected through this social network and obtain valuable feedback.

There are several reasons for this choice. Firstly, Facebook is used by users to create events and invite their friends to join activities, so our system can help them in the organization of such events. Secondly, users' activities in the social network can be tracked to obtain information about their trust with other users. And finally, it is a perfect environment to obtain users' social factors required by

[2] Trust values always are greater than 0 so we do not have problems with this normalization

[3] http://www.happymovie.net

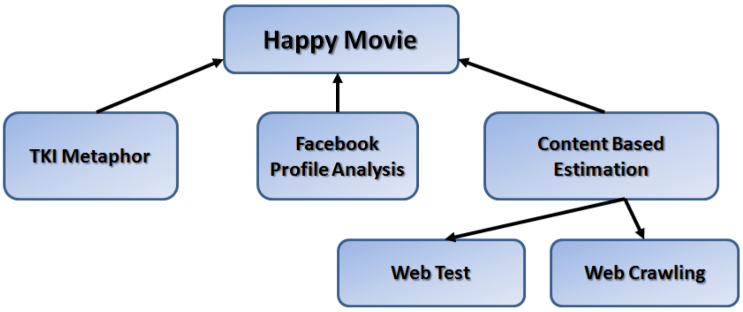


**Fig. 3** Overview of *HappyMovie*

our model as it is user-friendly, easily accesible, it has a lot of daily users and it's adapted to run questionnaires, applications and games.

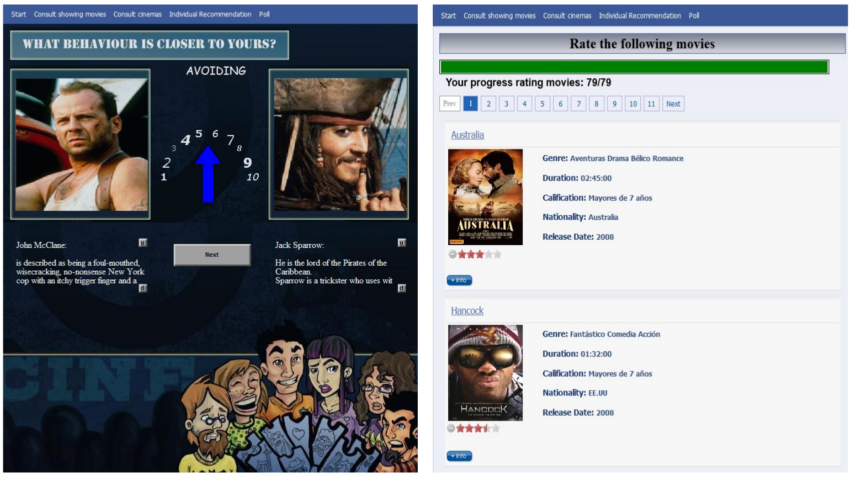
*HappyMovie*'s architecture is depicted in Figure 3. It is easy to compare it with the generic design of the ARISE architecture (Figure 1) described in Section 2. Next we summarize the way we have implemented the generic modules in our concrete system whereas Section 4.2 presents a functional overview of *HappyMovie*.

### 4.1 *HappyMovie* modules

*HappyMovie* instantiates the generic architecture of ARISE through the following modules (Figure 3):

**Cooperation, TKI Metaphor**: There are different approaches that can be used in order to obtain the different personalities or roles that people play when interacting in a decision making process. In our previous studies [44, 41, 42] we used the Thomas-Kilmann test [48]. We chose this test because is the most commonly used in the human-machine interaction area, due to its efficiency and that it is easy to evaluate and use for people not related to the psychology area. It provides a tangible and measurable value easy to interpret as opposed to other similar tests. According to this test, we can describe an individual's behaviour along two basic dimensions in conflict situations: (1) *assertiveness*, the extent to which a person attempts to satisfy her own concerns, and (2) *cooperativeness*, the extent to which a person attempts to satisfy other people's concerns. These two basic dimensions of behaviour define five different modes of responding to conflict situations: Competing, Accommodating, Avoiding, Collaborating and Compromising.

Our approach combines these 5 modes to obtain a personality value, $p_u$, representing the user's personality. To obtain the score that the user has in each mode, the TKI personality test proposes 30 situations where the user has to think about how s/he will react. Initially we used the original TKI test. However, when we asked our users about it, they described the test as tedious, long and not very clear in some of the questions. To make the application more easy to get through, in [43] we studied and validated the use of an alternative method to obtain data about a certain user's personality. It consists of an interactive metaphor that displays two well known movie characters with opposite personalities for each of five possible categories. One character represents the essential characteristics of one



John McClane:

is described as being a foul-mouthed, wisecracking, no-nonsense New York cop with an itchy trigger finger and a

Jack Sparrow:

He is the lord of the Pirates of the Caribbean.
Sparrow is a trickster who uses wit

Rate the following movies

Your progress rating movies: 79/79

Australia

Genre: Aventuras Drama Bélico Romance

Duration: 02:45:00

Calification: Mayores de 7 años

Nationality: Australia

Release Date: 2008

Hancock

Genre: Fantástico Comedia Acción

Duration: 01:32:00

Calification: Mayores de 7 años

Nationality: EE.UU

Release Date: 2008

(a) Personality test in *HappyMovie* (b) Preference test in *HappyMovie*

**Fig. 4** User tests in *HappyMovie*

category, while the other one represents all the opposite ones. What the user has to do is to choose, using a moving arrow, with which of each pair of characters s/he feels more identified. We have performed an experiment with real users using both tests, and proven that it is possible to replace the original TKI test with the new one (the metaphor) because the results obtained with the two tests are equitable (see [43] for the details of the experiment). In Figure 4(a) we can see how the personality test is presented in the application.

**Trust, Facebook Profile Analysis**: The Trust Module is the module that receives the largest benefit because the application is embedded in a social network. We are able to calculate the trust between users by extracting specific information from each of their own profiles in the social network. Facebook users usually post a huge amount of personal information that can be analysed to compute the trust in other users: distance in the social network, number of comments shared, likes and interests, personal information, pictures, games, duration of friendships, etc [19, 18].

In order to swift from theory to practice it is important to take into account that these elements are not easy to quantify and that obtaining them is limited by the extraction power that Facebook APIs give us. In *HappyMovie* we analyse the following factors: common friends, pictures in common, common interests (music, movies, series..) and comments on each other's Facebook walls. Afterwards, these factors are combined using a weighted average. We have adjusted the weights of these factors when calculating trust after an experiment with real users where they indicated the real trust that they had in each other. The trust between pairs of users is computed every time a user joins an active group (or in terms of *HappyMovie*, when a user joins an event related to a cinema outing). This calculation is done between the active user and the rest of the group members. For each pair of users and each event the trust value is only computed once. However, we do compute it again for each new event as Facebook profiles keep changing and so does the trust between two people. A detailed explanation of the trust factors obtained from Facebook and the combination process is provided in [42].

**Individual prediction**: Our group recommendation strategies combine individual predictions to find an item (movie) suitable for the group. This individual prediction module is built using the jCOLIBRI framework [15] and follows a content-based approach [40] to estimate the ratings a user would assign to each product in the catalogue. It compares the description of each product in the catalogue and selects those ones that are most similar to the user's preferences, and therefore, have the highest estimated rating. We have chosen a content-based system and not a collaborative one [16] because the movies to be recommended (i, in Figure 2) are too recent to have enough user ratings. Therefore, we could not use those ratings as collaborative recommenders do.

Consequently, this module has two requirements that must be fulfilled: the catalogue of products to be recommended, and the individual preferences of each user. In *HappyMovie* we obtain them with two sub-modules: a web crawling module that obtains new movie listings directly from the web and a web test module that obtains users' preferences.

To obtain the catalogue of products we have implemented a web crawler that obtains new movie listings from the web. This module is executed off-line and creates a data base of movies being played in cinemas. This data base also contains information about the location of the cinemas, the description of the movies and any other data required by our system.

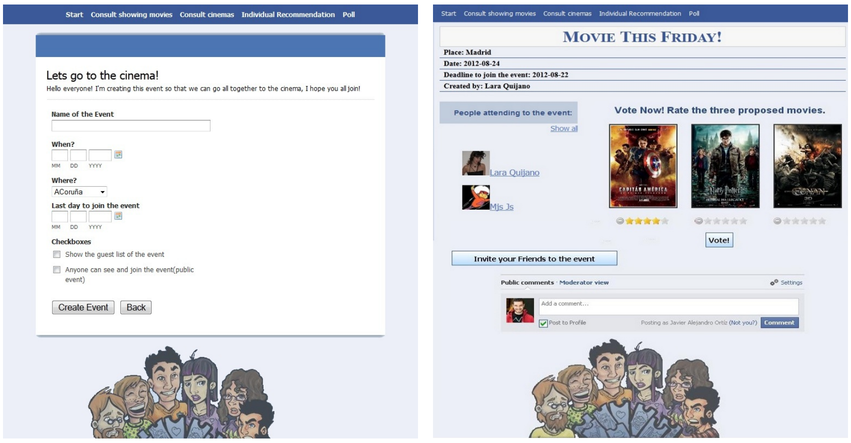
The web test module is in charge of obtaining users' preferences for movies. It consists of a test where users are provided with a set of heterogeneous movies that they should rate (20 at least) in a Likert scale from 0 to 5, as shown in Figure 4(b). This test must be run before using the *HappyMovie* application although it can also be run on demand to increase the accuracy of the system. These preferences will be later used to evaluate the satisfaction of users regarding the items proposed to the group.

Having described the implementation of the *HappyMovie* application we will briefly detail its behaviour to let readers understand its functionality.

### 4.2 Using the *HappyMovie* system

The necessary steps to obtain a movie group recommendation with *HappyMovie* are:

1. **Prerequisites** Before any user can access the movie recommendation functionality we collect the individual information required by our recommendation method. As we have previously explained, this information is the user's personality, trust and individual preferences.
   - In order for us to gather the necessary data about the user's personality, he or she will be made to choose among a set of characters the one they feel the most identified with (cooperation module), as shown in Figure 4(a).
   - In order to store information about users' preferences, he or she will be made to rate a set of movies (at least 20 movies), where they enter their personal preferences (this is the web test used by the individual prediction module), as shown in Figure 4(b).



Start Consult showing movies Consult cinemas Individual Recommendation Poll
Create Event
Back
Start Consult showing movies Consult cinemas Individual Recommendation Poll
Show all
Vote!
Invite your Friends to the event
Public comments · Moderator view
Settings
Comment

Lets go to the cinema!

Hello everyone! I'm creating this event so that we can go all together to the cinema, I hope you all join!

Name of the Event

When?
MM DD YYYY

Where?
ACoruña

Last day to join the event
MM DD YYYY

Checkboxes

□ Show the guest list of the event

□ Anyone can see and join the event(public event)

MOVIE THIS FRIDAY!

Place: Madrid
Date: 2012-08-24
Deadline to join the event: 2012-08-22
Created by: Lara Quijano

People attending to the event:

Lara Quijano
Mjs Js

Vote Now! Rate the three proposed movies.

Add a comment...

☑ Post to Profile

Posting as Javier Alejandro Ortíz (Not you?)

(a) Activity creation in *HappyMovie* (b) Group recommendation in *HappyMovie*

**Fig. 5** Events in *HappyMovie*

- In order to understand the user's circles of trust the application reads the information stored in the Facebook personal profile. It calculates the trust that the active user has in all the other users that have joined the event up to now.

2. **Activity definition**. *HappyMovie* identifies two different user roles. `Organizers` create the events as shown in Figure 5(a) and define the place, date or invited people. `Attenders` accept the invitation (delivered thought the Facebook capabilities) and can see the movies proposed by the system based on the current configuration of the group. As attenders they can invite further users or withdraw from the event: recommendations are proposed dynamically.
3. **Final choice.** Once the deadline is reached, the system recommends the (estimated) three best movies for the group. At this point they are allowed to rate each movie individually. This process lets them decide which movie they are finally going to watch and, more importantly, it gives the system the feedback required to evaluate the level of satisfaction of the group.

In next section we present an evaluation of our system where we compare the performance of the *delegation-based* prediction method, all the different aggregation functions and the impact of social factors.

## 5 Experimental evaluation

We have evaluated our movie recommendation method making use of *HappyMovie*. We have firstly performed our experiment with groups of real users and secondly repeated it with synthetic data. The most important goal of our experiment is to measure the performance of our group recommendation method in a real scenario. However we have decided to experiment also with synthetic data in order to explore extreme cases that could appear in conflict situations. We also want to have control of the data distribution, an impossible situation when using real data. This synthetic data lets us explore every group composition and personality distribution within a group. It also lets us reproduce the behaviour of large groups that are very difficult to organize in experiments with real users. In [42] we used this very same method to create synthetically generated data and proved that the results obtained were valid and equivalent to the ones obtained with real data.

The other goals of this experiment are:

- Prove that the delegation-based prediction method has a higher performance than the standard non-social group recommendation approach.
- Study which of the possible aggregation functions reports the highest performance with the *dbp* method.
- Analyse the impact of the personality and trust factors in the *dbp* method.

### 5.1 Experimental set-up

We developed a configurable group recommender implementing the aggregation strategies described in Section 3.1. This recommender can be configured to use the standard individual predictions returned by the individual estimations module, $pred(u, i)$, or our social-based prediction method, $dbp(u, i)$.

The inputs of the recommender are those ones defined by the modules in ARISE: personality $p_u$, trust $t_u$, and individual preferences $pref_u$ for each user $u$. The output is a set of items recommended for a given group configuration $rec(G)$. Finally, the validation data is the real group choice: an ordered list with the favourite items that the group would actually have chosen $fav(G)$. The size of both lists was limited to 3 items assuming that it is the maximum number of movies that a user/group would be really interested in watching at a time. The accuracy of the system will be measured by comparing $rec(G)$ and $fav(G)$. The more the recommended movie list resembles the real one, the better results our application is providing. The evaluation metrics applied to compare both sets are explained in Section 5.2.

Each configuration of the recommender was evaluated with two different input datasets. The first one was obtained from real users and the second dataset contains information from synthetically generated users. This artificial dataset let us explore the behaviour of the recommender with extreme or unusual group configurations. For example, we analysed the range of personality values for the real dataset and almost every user had a mild personality. Therefore we could not conclude whether or not our method performs accurately with extreme personalities. Additionally, the real-users dataset has a limitation regarding the size of the groups. To study the performance with large groups (over 10 members) we needed a considerable number of participants. The synthetic dataset solved this limitation and reproduced such an amount of users. We must note that the validity of this dataset has been already proven in our previous studies [42]. Next we describe the features of each dataset:

- **Real dataset:** As we mentioned above, the most important goal of our experiment was to measure the performance of our group recommendation method in a real scenario. To do so we used our Facebook application *HappyMovie*. We created different events in the social network as explained in Section 4 and asked volunteers to use it. The demographic data about our participants (mean age, gender, etc.) was quite varied because they were selected among colleagues and students. 58 users participated in our experiment. Input values $p_u$ and $pref_u$ were obtained from the tests presented in Section 4 (Figures 4(a) and 4(b)). The last input $t_u$ was obtained by analysing users Facebook profiles. The validation data $fav(G)$ was obtained by putting together groups of users that simulated going to the cinema together and gave us a 3 item list that contained the group choice. We managed to gather 15 groups of 9, 5 and 3 members (4, 6 and 5 groups respectively). To obtain the output list $rec(G)$, users created events in *HappyMovie* and joined them with the same configuration as they did in the simulation.

- **Synthetic dataset:** This second dataset lets us explore unusual group configurations. By using this approach we were able to group users in sets of 3, 5, 10, 15, 20 and 40 people.
  Personality $p_u$ is assigned randomly but following certain restrictions to ensure that we obtain groups composed of people with all the possible combinations of extreme personalities (very selfish, selfish, tolerant, cooperative and very cooperative).
  In the end we had 76 groups (13 different distributions for each size, except for the 40-person group where we only had 11 combinations due to the similarity of personalities in such big groups). The second input variable is the individual preferences $pref_u$. This is a very delicate step that we have resolved by assigning profiles to each user. These profiles are generated from the Movielens data set [10] according to typical preferences about movies stratified according to their age, sex and likes.
  The last input, $t_u$, was assigned randomly to each simulated user according to the typical distribution of trust in a population. To obtain the validation data $fav(G)$ we asked our volunteers to estimate which movies each artificial group would have chosen. The recommended output list $rec(G)$ was computed by applying our recommendations algorithms to the input data.

### 5.2 Evaluation metrics

Our experiment requires an evaluation function to measure the accuracy of the group recommendation. To do so, we compared the results of our recommender system $rec(G)$ to the real preferences of the users $fav(G)$. However the choice of a suitable evaluation metric requires the consideration of several factors.

The first one is the length limitation in the $fav(G)$ list. Real users are only interested in a few movies they really want to watch and consequently we limited them to 3 elements. Therefore, we cannot use general measures like recall or precision. Secondly, $rec(G)$ is an unordered set because our recommender proposes three movies without any kind of ranking that are afterwards voted by the members of the group to make their decision. This feature discards several evaluation metrics that compare the ordering of the output and validation lists like the Mean Absolute Error (MAE) [22,2] or the Normalized Discounted Cumulative Gain (nDCG) [5].

However, there are some metrics used in the Information Extraction field [49] that are suitable for our scenario. In our case, we can use *precision@3* to evaluate how many of the movies in $rec(G)$ are in the $fav(G)$. This kind of evaluation can be seen from a different point of view: we are usually interested in having at least one of the movies from *rec*$(G)$ in the *fav*$(G)$ list. This measure is called *success@3* and returns 1 if there is at least one hit in the first *3* positions. Therefore, we could use *success*@3 (or simply $s$@3) to evaluate our system by computing the rate of recommendations where we have at least one hit in *fav*$(G)$. For example, 90% accuracy using $s$@3 represents that the recommender suggests at least one correct movie for 90% of the groups being evaluated. In fact, $s$@3 is equivalent to having *precision*@3 $> 1/3$. We can also define a $2s$@3 metric (equivalent to *precision*@3 $> 2/3$), which represents how many times *fav*$(G)$ contains at least two movies from *rec*$(G)$. Obviously, it is a much more restrictive measure.

### 5.3 Results

In this section we detail the results obtained with different configurations of the recommender. Each configuration is defined by the input dataset (real or synthetic), the aggregation function (from Section 3.1) and the estimation method: *pred*$(u, i)$ or *dbp*$(u, i)$. As we have have studied 7 aggregation strategies we have ended up with 28 different configurations. To simplify the reporting of the results we group these configurations into two sets according to the estimation method. Those configurations using the basic estimation *pred*$(u, i)$ without social knowledge conform the baseline of the results and will be referred to as "Base configuration". Complementarily, configurations using the *delegation-based* prediction method will be labeled as "dbp configuration".

First, we analysed the improvement of the dbp configuration with respect to the baseline for each aggregation function. In Figures 6 and 7 we can see the comparison of the results for the real data and the synthetic dataset. In average the improvement rates for the real dataset are 13.33% with $s$@3 and 3.8% with $2s$@3. For the synthetic dataset improvements are 15.22% and 14.84% for each evaluation measure respectively.

Next, we explored the correlation of these improvements to the social factors. Our delegation-based method integrates two different social factors: personality and trust. To figure out the impact of this social behaviour knowledge in the recommendation process we repeated the evaluation of the dbp configurations with three different flavours of the *dbp* method where each social factor was nullified: only personality $dbp_p$, only trust $dbp_t$ and no social knowledge at all $dbp_{null}$. This last variant let us measure the impact of the collaborative approach followed by the *dbp* method where individual preferences are predicted by averaging other users' preferences. Results are presented in Figure 8. As we can observe, these variants can achieve at the most the same results as the full dbr approach for several aggregation functions. However it is not possible to generalize these results because their accuracy varies depending on the aggregation strategy and the dataset being used. Therefore we cannot conclude that, in general, these variants can be used to maximize the performance of a global system that works with different group configurations. Nevertheless, Figure 8 illustrates a relevant finding: the full dbr method always performs at least as well as the other variants. It is the best way to balance the social factors included in our recommendations model: personality and trust. The statistical significance of these results was confirmed using the Wilcoxon signed-rank test ($p < 0.05$). Consequently, we can conclude that our

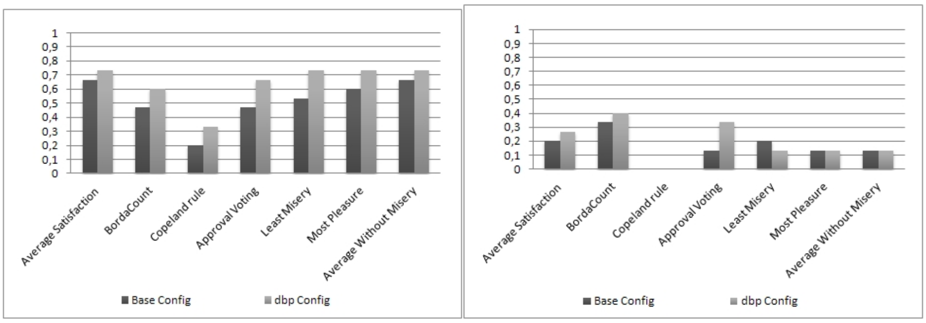


**Fig. 6** Performance of the base and dbp configurations w.r.t. each average function using the real dataset and the s@3 (left) and 2s@3 (right) evaluation metrics.

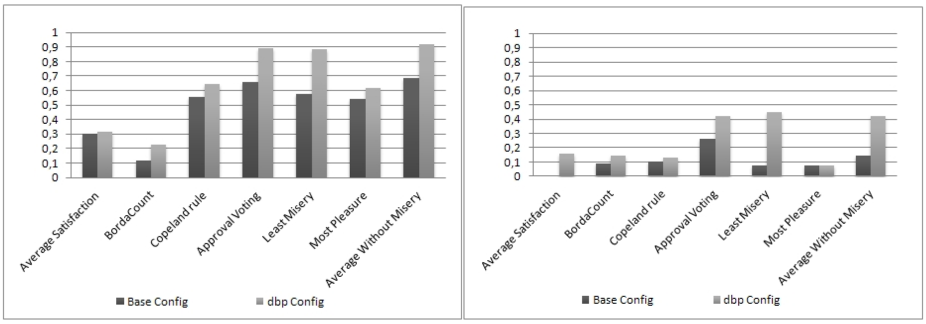


**Fig. 7** Performance of the base and dbp configurations w.r.t. each average function using the synthetic dataset and the s@3 (left) and 2s@3 (right) evaluation metrics.

delegation-based prediction method significantly improves standard aggregative recommendation approaches and performs at least as well as the other variants.

Finally, we analysed the performance of the delegation-based prediction method according to group size.

Figure 9 illustrates the results achieved by *dbp* applied to the real and synthetic datasets, for different group sizes and every aggregation function. Upon analysing the results from the real dataset (Figure 9 (left)) we can observe that while some aggregation functions such as *average satisfaction* report better results for small groups (we consider groups of 10 or less as small), others like *least misery*, *most pleasure* or *average without misery* work the other way round and obtain better results for large groups. This is the reason why we needed synthetically generated data: to study the specific aggregation strategy that each posible group size requires. However, we can conclude that, on average, the best aggregation function for such small groups is the *average satisfaction* strategy.

Figure 9 (right) shows the results for the synthetic dataset. Here we can confirm that *least misery* and *average without misery* are the optimal aggregation functions for large groups. Both show a rise in their performance when applied to a larger group. We can also observe that the *average without misery* strategy has decreasing performance although it reported good results for 9-member groups in the experiment with real data. When analysing the behaviour of the *average satisfaction* strategy we can confirm its unsatisfactory performance with large groups.

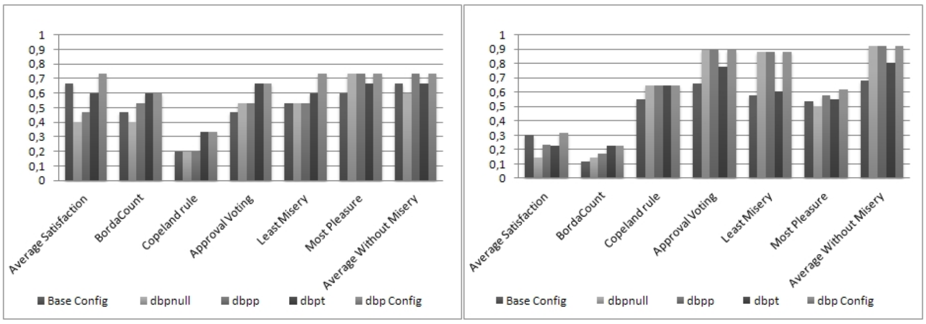


**Fig. 8** Performance of the dbp variants w.r.t. each average function using both datasets real (left) and synthetic (right) and the s@3 evaluation metric

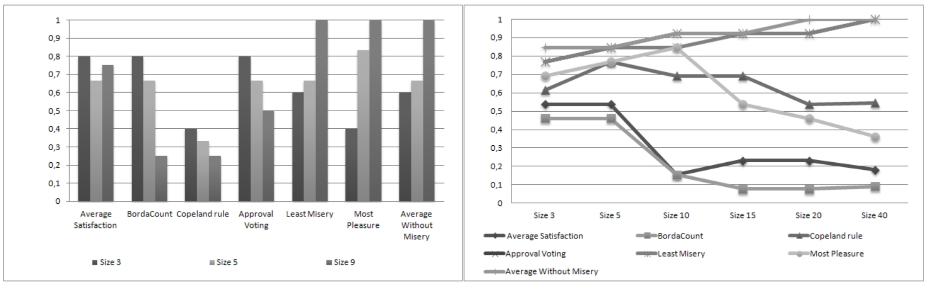


**Fig. 9** Performance of the dbp configuration and group sizes w.r.t. each average function using the real dataset and the s@3 evaluation metric.

A related work section will be introduced next explaining all the different works that have been done in the field of group recommenders and their differences with our system.

## 6 Related Work

There are a lot of domains where group recommendation techniques can be applied. For example, in the music domain, the work presented in [30] uses an algorithm that personalizes the distance measure between different pieces of music based on user preferences. MusicFX [35] provides recommendations about the background music at a fitness centre based on the preferences provided in different musical genres by the users. We can also find FlyTrap [13], a group recommender which selects music to be played in a public room. In the movies domain, Polylens [38] is an extension of Movielens to generate recommendations to groups. Regarding recommendations of restaurants for groups, we can find an interesting recommender system, Pocket Restaurant Finder [34], which bases its strategy on users' locations and the culinary characteristics of the restaurant. To find the best TV program to watch we have YuTV [51], which uses a vector space model with features of the TV programs (such as genre or actors) to find relevant recommendations for groups. LET'S BROWSE [29] is another example of group recommendation; this one recommends web pages to a group of two or more persons who are browsing the web together.

What all these recommenders have in common is that they take into account personal preferences obtained from their users; however, they consider each user to be equal to the others. The recommendation is not influenced by their personality or the way each one behaves in a group when joining a decision-making process. In our approach we propose to study how people interact depending on their personality or their closeness in order to improve group recommendations.

### 6.1 More than preference aggregations

There are also other works, besides our own, that do take into account not only the preferences of every member but also the interaction among them; Travel Forum Decision [23] is an example of this. The goal of this application is to help groups of users plan their vacations together. The system provides a solution and allows group members to discuss. It acts as a mediator until they reach a solution. In our approach, we propose to simulate this discussion in order to relieve our users of the process of discussing about choosing a solution. This way, our proposal requires less interaction from the users and presents an immediate solution.

When considering more than just individual preferences, there are some systems, such as CATS [36] which is a conversational recommender for planning skiing holidays, that take into account the attitudes and behaviour of other group members. Other systems that make a more detailed study of the group before making any recommendations are Intrigue [4], which plans visits for groups of tourists by weighting the preferences of different subgroups with special needs (like children or disabled people). Chen [11] proposes the use of genetic algorithms to learn group preferences by using the known preferences of the subgroups within a group. Although the results seem to be significant, they suppose that groups are fixed and they have previously rated some items together. In the case of CATS, the recommendation is defined as an incremental process where users collaboratively refine the suggested recommendation by critiquing its features or discarding it. They consider that the preferences of the current member partially depends on the preferences and/or the anticipated behaviour of other members. During the process of choosing a recommendation, users can see what other members have voted for, so they are conditioned by other members's opinions. CATS users need to read the information of other users in order to alter their initial opinion. Obviously this is only possible for users who vote later. Our approach simulates this conditioning more thoroughly, because it can simulate these alterations beforehand by taking into account the strength of the relationships between group members.

Other works focus on the integration of group disagreements in the recommendation process. One of the most recent systems is GRec-OC [25], a book recommender system for online communities. GRec-OC provides recommendations based on the books that other similar groups have purchased and tries to reduce the dissatisfaction of individual members. The work in [3] proposes a recommender that aggregates prior group member preferences to create the recommendation. Then, preference disagreements between pairs of individuals are collected and employed to score and rank the recommended items. Finally, Masthoff and Gatt [33] use individual satisfaction and emotional contagion in order to recommend a sequence of video clips for a group. The authors think that a member changes the selection of her/his best clip according to the clip selected during the previous selection step. This change can be reflected in the recommendation algorithm as an individual satisfaction function that computes the individual affective state. This state influences the affective state of the other members, producing an emotional contagion that should be taken into account during the recommendation process. Additionally, they point out a tendency in the influence of social status on the selection process.

Summing up, we conclude that there is a need to adapt the recommendation process to group composition [24,32]. This is backed up by some recent works that have focused their studies on analyzing the effectiveness of group recommendations according to different aspects, such as group size and inner group similarity [5], or on studying different weighting models to combine the preferences of group members according to their activity or role within the group [8]. Additionally, it is also known that a user's preferences can be affected by the rest of the group [32,11]. However, most of the aggregation strategies employed in previous works combine users' preferences without taking into account either the relationships between group members or the relevance of each member's preferences. The work dealing with these issues is limited. We observed that there was a need to modify those existing strategies that consider each user of the group to be equal to the others. So we focused our line of work on reflecting each user's individual aspects and how they interact with each other.

In [44] we presented an improvement of current group recommendation techniques by introducing a novel factor: the personality of each individual in the group when dealing with conflict situations. We use a personality test to obtain the different roles that people play when interacting in a decision-making process. Once we studied the individual characterization of people in a group, we decided to study other factors regarding the structure of the group itself and how users interact with each other. The inclusion of the individual personality factor wasn't enough to achieve this, due to the increasing importance of social networks and the trust connections that they imply. Therefore we needed to explore more social factors.

### 6.2 Social recommendations in social networks

In the last few years researchers have proved that the inclusion of social aspects in the recommendation processes improves the recommendation accuracy [14]. Social networks such as Facebook or Twitter can provide a rich mine of resources and the possibility of acquiring data about the user's circles of trust. These networks contain implicit information that can be used in a recommendation process [27]. This option, which is completely transparent to users, has as a main advantage that users are not required to provide explicit information about their trust in other users. This information is extracted implicitly from their daily interaction in the social network. However, it has the obvious drawback that every user involved in the group recommendation process must belong to the social network. Nevertheless, the rising popularity of this kind of web applications minimizes this risk. Even more, it is becoming usual to organize events (such as going to the cinema) through social networks, so group recommendation techniques could be integrated into these web sites. In [19] Golbeck proposes a methodology to infer relationships of trust within social networks. The computational problem of trust is to determine how much one person in the network should trust another. Certainly, trust inferences will not be as accurate as a direct rating. But the algorithm presented in this study, named TidalTrust, managed to improve the accuracy by 10%.

It is a fact that people rely more on recommendations from people they trust (friends) than on recommendations based on anonymous ratings [47]. This is a very important factor in group recommendation strategies, when a decision for the whole group has to be made. This kind of recommendations usually follow an argumentation process, where each user defends her preferences and rebuts others' opinions. Here, the tie strength or trust between users is crucial because they must adjust their opinions in order to reach a common decision.

The generation of trust models has created a huge body of work. The emergence of the current collaborative web (Web 2.0) has boosted the idea of the Web Of Trust (WOT) [20,39,50]. The WOT represents the trust between users, modeled using an online network. There are specific approaches that use a custom trust network to recommend items. One example is FilmTrust [20], which exploits a custom network of trust between users regarding movie preferences. However, these specific trust networks are quite difficult to generate because they require explicit feedback from users, and this can generate rejection.

All these works take into account some of the different factors involved in our proposal: personality or trust. However, we have not found any work that integrates and evaluates these two factors in group recommendation processes. Therefore, we consider that our approach improves these works by making a more exact representation of how group argumentations take place in real life.

## 7 Conclusions

In this paper we have reviewed existing techniques in group recommender systems and contribute to the state of the art with a method of making group recommendations that includes social elements. The paper describes the ARISE architecture for the development of group recommenders that takes into account social factors like trust, conformity and fairness. The inclusion of these factors leads to a significant improvement in the performance of the recommendations. ARISE is a theoretical organization of the modules required to build such kind of enhanced recommenders, which has been instantiated in the *HappyMovie* application. *HappyMovie* is a real application that serves as a proof of concept and was developed to exploit Facebook in order to obtain social information about users.

The main focus in this paper is the study of how to apply the methods proposed in ARISE -exploiting information about the social relationships and behaviour of the users to provide better recommendations- to a wide range of different aggregation functions that help us combine all the information extracted from users in order to build the final group recommendation. We have tested the behaviour of our recommender with all its possible configurations in the movie recommendation domain using two test datasets. The first case study uses real users and the second one uses synthetically generated data to create simulated groups of people.

In both experiments we have used groups of different sizes and personal preferences, and different aggregation functions, where we have proved that by introducing the trust factor and personality awareness we improve the results of the recommendations. We have also studied several features of group composition to measure their impact on the accuracy of the group recommender. Regarding the influence of group size, the conclusion is that we obtain better results for small groups with *average satisfaction* and for big groups with *least misery* or *average without misery*.

These conclusions lead us to propose as future work an adaptive recommender that applies different aggregation functions depending on the group. This means that depending on the configuration of the group we will choose a different approach to compute the final group recommendation.

We are also working on the possibility of extending the group recommendation application for movies to other domains like music, recipes, trips and/or restaurants. This way we can validate our group recommendation method. To do so we would only have to improve the *Web Crawling Module* so that it searches the web for specific information on each domain and builds case bases. We will have to modify the *Individual Preferences Module* as well in order to ask users to rate items in the specific domain.

## 8 Acknowledgments

Supported by Spanish Ministry of Science & Education (TIN2009-13692-C03-03) and Madrid Education Council and UCM (Group 910494) and Spanish Ministry of Economy and Competitiveness under grant IPT-2011-1890-430000.

## References

1. Abramowitz, M., Stegun, I.A.: Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, ninth dover printing, tenth gpo printing edn. Dover (1964)
2. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. p. 734749 (2005)
3. Amer-Yahia, S., Roy, S.B., Chawlat, A., Das, G., Yu, C.: Group recommendation: semantics and efficiency. Proceedings of the VLDB Endowment **2**(1), 754–765 (2009)
4. Ardissono, L., Goy, A., Petrone, G., Segnan, M., Torasso, P.: Intrigue: Personalized recommendation of tourist attractions for desktop and handset devices. Applied Artificial Intelligence **17**(8), 687–714 (2003). URL http://dx.doi.org/10.1080/713827254
5. Baltrunas, L., Makcinskas, T., Ricci, F.: Group recommendations with rank aggregation and collaborative filtering. In: Proceedings of the fourth ACM conference on Recommender systems, RecSys '10, pp. 119–126. ACM, New York, NY, USA (2010). DOI 10.1145/1864708.1864733
6. Barsade, S.G.: The ripple effect: Emotional contagion and its influence on group behavior. Administrative Science Quarterly **47**(4), pp. 644–675 (2002)
7. Bazire, M., Brézillon, P.: Understanding context before using it. In: Context, pp. 29–40 (2005)
8. Berkovsky, S., Freyne, J.: Group-based recipe recommendations: analysis of data aggregation strategies. In: Proceedings of the fourth ACM conference on Recommender systems, RecSys '10, pp. 111–118. ACM, New York, NY, USA (2010). DOI 10.1145/1864708.1864732
9. Bischoff, K.: Exploiting social ties for search and recommendation in online social networks - challenges and chances. In: Grundlagen von Datenbanken (2010)
10. Bobadilla, J., Serradilla, F., Hernando, A.: Collaborative filtering adapted to recommender systems of e-learning. Knowl.-Based Syst. **22**(4), 261–265 (2009)

11. Chen, Y.L., Cheng, L.C., Chuang, C.N.: A group recommendation system with consideration of interactions among group members. Expert Syst. Appl. **34**(3), 2082–2090 (2008). DOI http://dx.doi.org/10.1016/j.eswa.2007.02.008
12. Christakis, N.A., Fowler, J.H.: Social contagion theory: Examining dynamic social networks and human behavior. CoRR (2011)
13. Crossen, A., Budzik, J., Hammond, K.J.: Flytrap: intelligent group music recommendation. In: IUI '02: Proceedings of the 7th international conference on Intelligent user interfaces, pp. 184–185. ACM (2002). DOI http://doi.acm.org/10.1145/502716.502748
14. Davoodi, E., Kianmehr, K., Afsharchi, M.: A semantic social network-based expert recommender system. Appl. Intell. **39**(1), 1–13 (2013)
15. 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**(1-3), 68–75 (2007)
16. Ekstrand, M.D., Riedl, J., Konstan, J.A.: Collaborative filtering recommender systems. Foundations and Trends in Human-Computer Interaction **4**(2), 175–243 (2011)
17. Erdélyi, G., Nowak, M., Rothe, J.: Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control. Math. Log. Q. **55**(4), 425–443 (2009)
18. Gilbert, E., Karahalios, K.: Predicting tie strength with social media. In: CHI '09: Proceedings of the 27th international conference on Human factors in computing systems, pp. 211–220. ACM (2009). DOI http://doi.acm.org/10.1145/1518701.1518736
19. Golbeck, J.: Combining provenance with trust in social networks for semantic web content filtering. In: L. Moreau, I.T. Foster (eds.) Provenance and Annotation of Data, International Provenance and Annotation Workshop, IPAW 2006, Chicago, IL, USA, May 3-5, 2006, Revised Selected Papers, *Lecture Notes in Computer Science*, vol. 4145, pp. 101–108. Springer (2006)
20. Golbeck, J.: Generating predictive movie recommendations from trust in social networks. In: iTrust: 4th International Conference on Trust Management, pp. 93–104 (2006)
21. Hatfield, E., Cacioppo, J., Rapson, R.: Emotional Contagion. Studies in Emotion and Social Interaction. Cambridge University Press (1994)
22. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. **22**(1), 5–53 (2004). DOI http://doi.acm.org/10.1145/963770.963772
23. Jameson, A.: More than the sum of its members: challenges for group recommender systems. In: AVI '04: Proceedings of the working conference on Advanced visual interfaces, pp. 48–54. ACM (2004). DOI http://doi.acm.org/10.1145/989863.989869
24. Jameson, A., Smyth, B.: Recommendation to groups. In: P. Brusilovsky, A. Kobsa, W. Nejdl (eds.) The Adaptive Web, Methods and Strategies of Web Personalization, *Lecture Notes in Computer Science*, vol. 4321, pp. 596–627. Springer (2007)
25. Kim, J.K., Kim, H.K., Oh, H.Y., Ryu, Y.U.: A group recommendation system for online communities. International Journal of Information Management **30**(3), 212 – 219 (2010). DOI DOI: 10.1016/j.ijinfomgt.2009.09.006. URL http://www.sciencedirect.com/science/article/B6VB4-4XHJX9H-4/2/29b4c6d7fb829946dc84b9db4bcab5a2
26. Klamler, C.: The copeland rule and condorcet's principle. Economic Theory **25**(3), 745–749 (2005)
27. Koohborfardhaghighi, S., Kim, J.: Using structural information for distributed recommendation in a social network. Appl. Intell. **38**(2), 255–266 (2013)
28. Levin, D.Z., Cross, R., Abrams, L.C.: The strength of weak ties you can trust: the mediating role of trust in effective knowledge transfer. Management Science **50**, 1477–1490 (2004)
29. Lieberman, H., Dyke, N.W.V., Vivacqua, A.S.: Let's browse: A collaborative web browsing agent. In: IUI, pp. 65–68 (1999)
30. Liu, N.H.: Comparison of content-based music recommendation using different distance estimation methods. Appl. Intell. **38**(2), 160–174 (2013)
31. Lops, P., de Gemmis, M., Semeraro, G.: Content-based recommender systems: State of the art and trends. In: Ricci et al. [45], pp. 73–105
32. Masthoff, J.: Group modeling: Selecting a sequence of television items to suit a group of viewers. User Modeling and User-Adapted Interaction **14**(1), 37–85 (2004)
33. 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**(3-4), 281–319 (2006)

34. McCarthy, J.F.: Pocket restaurant finder: A situated recommender systems for groups. In: Proceeding of Workshop on Mobile Ad-Hoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems (2002)
35. McCarthy, J.F., Anagnost, T.D.: MusicFX: An arbiter of group preferences for computer aupported collaborative workouts. In: CSCW '98: Proceedings of the 1998 ACM conference on Computer supported cooperative work, pp. 363–372. ACM (1998). DOI http://doi.acm.org/10.1145/289444.289511
36. McCarthy, K., McGinty, L., Smyth, B., Salamó, M.: The needs of the many: A case-based group recommender system. Advances in Case-Based Reasoning **4106**, 196–210 (2006). DOI 10.1007/11805816_16. URL http://dx.doi.org/10.1007/11805816_16
37. Mika, P.: Making things findable: semantics for web search and online media. In: WIMS, p. 3 (2011)
38. O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J.: Polylens: a recommender system for groups of users. In: ECSCW'01: Proceedings of the seventh conference on European Conference on Computer Supported Cooperative Work, pp. 199–218. Kluwer Academic Publishers, Norwell, MA, USA (2001)
39. O'Donovan, J., Smyth, B.: Trust in recommender systems. In: IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces, pp. 167–174. ACM (2005). DOI http://doi.acm.org/10.1145/1040830.1040870
40. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: The Adaptive Web, pp. 325–341 (2007)
41. 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)
42. 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'10., pp. 121–126. IEEE Computing Society (2010)
43. Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B.: Happymovie: A facebook application for recommending movies to groups. In: Procs of the 23th International Conference on Tools with Artificial Intelligence, ICTAI'11. (2011)
44. Recio-García, J.A., Jimenez-Diaz, G., Sánchez-Ruiz, A.A., Díaz-Agudo, B.: Personality aware recommendations to groups. In: Procs. of the 2009 ACM Conference on Recommender Systems, pp. 325–328. ACM (2009)
45. Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.): Recommender Systems Handbook. Springer (2011)
46. Saari, D.G.: Mathematical structure of voting paradoxes. Economic Theory **15**(1), 1–53 (2000)
47. Sinha, R.R., Swearingen, K.: Comparing recommendations made by online systems and friends. In: DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries (2001)
48. Thomas, K., Kilmann, R.: Thomas-Kilmann Conflict Mode Instrument. Tuxedo, N.Y. (1974)
49. Tomlinson, S.: Comparing the robustness of expansion techniques and retrieval measures. In: CLEF, pp. 129–136 (2006)
50. Victor, P., Cornelis, C., Cock, M.D., Teredesai, A.: Key figure impact in trust-enhanced recommender systems. AI Commun. **21**(2-3), 127–143 (2008)
51. Yu, Z., Zhou, X., Hao, Y., Gu, J.: TV program recommendation for multiple viewers based on user profile merging. User Modeling and User-Adapted Interaction **16**(1), 63–82 (2006). DOI http://dx.doi.org/10.1007/s11257-006-9005-6