Using Personality to Create Alliances in Group Recommender Systems*

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

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

Abstract. Our recent work analyses the accuracy of group recommenders when using information about the personality and the social connections between the members of the group. The goal in this paper is the use of personality and trust as the mean to define alliances to reach agreements inside a group of people. The approach reproduces the behaviour of real users when negotiating a common item to consume using three variables: personality, trust and personal preferences. We run an experiment in the movie recommendation domain where we use a personality test to identify the group leaders and test the number of people they are able to convince about a certain item to consume.

1 Introduction

Recommender systems have been one of the main application areas of the techniques commonly used in the Case-Based Reasoning field [1,2]. The analogies between Case Based Reasoning (CBR) and recommenders are obvious. Recommender systems manage items instead of cases but the retrieval methods are very similar. Once the best item is obtained it is proposed directly to the user without requiring adaptation. Moreover, both techniques pay an important attention to the learning processes that improve the performance of the systems by taking into account the preferences or experiences of the users. In a general way we could apply two different approaches. Collaborative recommenders use the ratings already assigned by the 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 side, the content-based approach compares each item to be proposed with the items already rated by the target user. Then the ratings of the most similar rated items are combined to provide an estimation.

Our recent work [3,4,5,6] analyses the accuracy of group recommenders when using information about the personality and the the social connections between the members of the group. Typically a group recommender uses several subsets

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

A. Ram and N. Wiratunga (Eds.): ICCBR 2011, LNAI 6880, pp. 226–240, 2011.

of preferences -one per person- that are combined to create a global recommendation suitable for everyone in the group. Simpler existing works on group recommender systems are based on the aggregation of the preferences of every member of the group, where each member is considered with the same degree of importance [7,8]. However, groups of people can have very different characteristics like size and can be made of people with similar or antagonistic personal preferences. It is a fact that when we face a situation in which the concerns of people appear to be incompatible, a conflict situation arises.

Our previous approaches determine that the general satisfaction of the group is not always the aggregation of the satisfaction of its members as different people have different expectations and behaviour in conflict situations. The *personality* factor reflects the cooperativeness or selfishness of each user when selecting a product for the whole group. This fact is taken into account in recent works that agree on the need of adapting the recommendation process to the group composition. Furthermore, it is also known that the user preferences can be affected by other people of the group and can change over the time [9,2,10]. Personality allows us measuring the degree of acceptance of the products proposed by other users and the way of solving conflicts. Our research characterizes people using the Thomas-Kilmann Conflict Mode Instrument (TKI) [11] that describes a person behavior in conflict situations.

The concept of *trust* [12], can be defined as the extent to which one party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible. Trust networks consist of transitive trust relationships between people, organizations and software agents connected through a medium for communication and interaction. 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 [13].

In this paper we describe a new approach to solve conflict situations by modeling users interaction in group recommender systems. Instead of computing a global recommendation for the group of people based on the individual preferences and personality of its members, we propose a model where each user negotiates to convince other members about a common item to consume. We exploit the principle of *homophily*, people that share interest with their friends and tend to be friends with people who share their interests. This feature has been shown to exist in many social networks [14,15]. In our model, users with strong personalities try to create *alliances* with other users to support their personal preferences. This way, influencer users obtain the required votes to get their proposal chosen by the group. These influencers, or leaders, try to influence other users and they use their leadership to create the alliance.

Influencers, are typically characterized as thought leaders, or just plain interesting personalities who have the ability to influence potential users. In practice, these individuals may be identified as highly connected individuals or individuals that bridge (also called *connectors* [16]) two relatively large sub-communities. This social behaviour has been extensively researched in the social sciences over the past few decades [17],[14],[18].

Our new approach uses personality and trust as the mean to define *alliances* inside a group of people. An alliance is defined as a subgroup that agree about the same recommendation result. A leader creates alliances with other users (s)he trusts in order to support a concrete product p . The product in the alliance with the bigger number of members is chosen as the global recommendation result. A total agreement situation leads to an alliance including all the people of the group.

Summing up, in this paper we propose a model based on alliances to provide recommendations to groups. We identify leaders by a personality test. Potential allies are obtained by computing the trust between users. Leaders negotiate with their closer friends to conform an alliance that has the majority of votes required to get the influencer's favourite items.

The paper runs as follows. Section2 introduces related work. In Section 3 and 4 we explain an overview of our previous research, a generic architecture for group recommendations, ARISE, that uses personality and trust values in order to improve group recommendations. Section 5 describes the method based on alliances that we propose in this paper. Section 6 describes a case study in the movie recommendation domain and presents some results on the use of alliances in the group decision making. Section 7 concludes the paper.

2 Related Work

Related works about creating alliances and the role of influencers are shown in some online social communities. A *coalition* from social agents area is defined as a temporary association between agents in order to carry out joint projects. The aim is to achieve complex projects by using a better distribution of competencies. An example is the approach of [19] to solve a cooperative game. Different works study automatic methods for coalition formation [20] or properties like efficiency, optimality or stability of the coalition structure [21,22]. Our approach is also related with voting games [23], a popular model of collaboration in multiagent systems. In such games, each agent has a weight (intuitively corresponding to resources he can contribute), and a coalition of agents wins if its total weight meets or exceeds a given threshold.

Our theory is based on the idea of a distributed group recommender system based on previous research on distributed Case Based Reasoning. Distributed CBR assumes multi-case base architectures involving multiple processing agents differing in their problem solving experiences [24]. In this new scenario each case base contains a list of contents, like products, rated by the user. These ratings represent the users explicit preferences that belong to the user model. These individual ratings are later combined with the ratings from other users to obtain a joint recommendation for the group. CBR literature proposes several ways to combine several experiences to obtain improved solutions in distributed architectures. One important method is the *ensemble effect* explained in [25] which proves that the argumentation of two agents improves the results obtained by one only agent working with the same experiences. This conclusion was the precursor of a research line focused on finding the best argumentation protocols to allow CBR agents to discuss about a common problem. In [25] they came up with the AMAL protocol that enables several CBR agents to argue about a common problem by means arguments and counterarguments. We have adapted the idea of agents giving arguments to validate their proposal, to an approach where the agents are influencers who give arguments to try to convince other users they are close to, to support their proposal.

The motivation and main contribution of this work is to use the ideas of alliances formation and collaboration between agents to improve group recommender systems. However in our model people of the same alliance do not collaborate to solve a complex project but reach an agreement on the item to be consumed by the whole group. So, our model does not represent knowledge about agent competencies or resources to contribute. It represents information about people's preferences, personality and trust that are used to convince the other members in the group.

In our method, leaders, who we call influencers, try to wield influence over friends to achieve their own goals. This must be taken into account when recommending items to groups of friends. The main problem when applying this model is the identification of potential influencers and influenced friends. However social networks provide (partially) these data. We can compute the trust between users to measure the closeness of their relationship and therefore the possibility of influence. However, social connections aren't enough for identifying influencers. To do so, we propose to measure the personality of the users.

3 ARISE: Generic Architecture for Group Recommenders Using Social Elements

Our approach, presented in [4,5,6] determines that the general satisfaction of the group is not always the aggregation of the satisfaction of its members, as groups of people can have very different characteristics. The inclusion of social elements into a group recommendation strategy is what we call ARISE[1] (Figure 1). This architecture allows us to simulate in a more realistic way the decision process followed by groups of people when choosing a joint activity.

The architecture of ARISE [6] is divided in six different modules: personality, trust, memory and satisfaction individual preferences estimation, explicit individual preferences, and product data. The information provided by each module is combined by the ARISE's group recommendation methods described in Section 4. Next, we summarize modules functionality:

- **Personality Module.** When making group decision processes there are situations where the concerns of people appear to be incompatible and *conflict situation* arises. Different people have different expectations and behaviour in conflict situations that should be taken into account. We have studied the different behaviours that people have in conflict situations according to their

[1] ARISE stands for Architecture for Recommenders Including Social Elements.

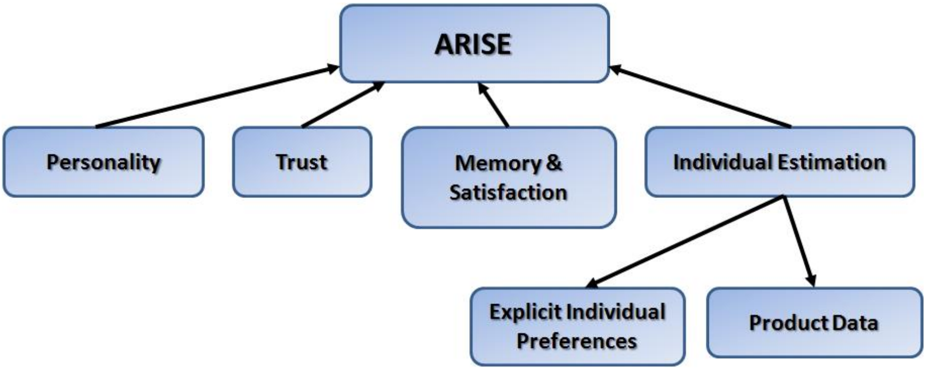

Fig. 1. Facebook application architecture: ARISE

personality. Personality module fulfils the task of obtaining a value that represents the personality of each user. This value, p, is a number $\epsilon[0, 1]$ where 1 represents a very strong personality and 0 a very easy going personality. In the ARISE architecture it is described as a high-level module that can be implemented in different ways. We obtain this factor using a popular personality test called TKI [11]. We have chosen this test because it takes very little time to answer it and the questions about the users personality are asked in an indirect way, not digging into too personal questions. In that way users do not resent from a excessively tedious test to answer.

- **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 [26]. In this module we evaluate information stored in our users profiles inside a social network, Facebook. With this information we compute the trust between users. Examples of these *social factors* are distance in the social network, number of common friends, intensity, intimacy or duration of the relationship.

  The details of the trust and the personality computation are fully detailed in [4,5].
- **Memory and Satisfaction Module.** After applying the personality and trust factors we must assure a certain degree of satisfaction between all the members of the group. We propose the use of a memory of past recommendations. Having recommendations with memory means that we are able to create a system that remembers all the previous recommendations for a given group. We believe that this is a necessary step when providing a whole set of fair recommendations.
- **Individual Preferences Estimation.** Our recommendation strategies predict the rating that each user would assign to every item in the catalogue and then these estimated ratings are combined to obtain a global prediction for the group. Finally, the product with the highest prediction is proposed. Therefore, a basic building block of the architecture is the module in charge

of the computation of the individual predictions. For the construction of the individual recommender we use the jCOLIBRI framework [27]. jCOLIBRI is currently a reference platform in the Case-Based Reasoning (CBR) community that facilities the design of different types of CBR applications and it has a specific extension for developing recommender systems.

Independently of the approach chosen to implement this generic module of the ARISE's architecture, there are two components (or submodules) that are always required by the individual recommender: A) the explicit individual preferences, which spans any kind of information about the user that is required to predict the rating for a new item. Commonly, it just consists on the ratings given to some products in the catalogue. B) the product data set, which provides the information about the items in the catalogue that should be recommended to the group.

4 Group Recommendation Methods in ARISE

Our group recommendation method is based on the typical preference aggregation approaches. These approaches [7,8] 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, this group recommendation method is what we call a base group recommender.

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

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 and trust factors. This way, we modify the impact of the individual preferences as shown in Equation 2.

$$\begin{aligned} gpred(G,i) &= \bigsqcup_{\forall u \in G} pred'(u,i) \\ pred'(u,i) &= \bigsqcup_{\forall v \in G} f(\, pred(u,i)\, ,\, p_u\, ,\, t_{u,v}\,) \end{aligned} \tag{2}$$

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 and $t_{u,v}$ is the trust value between users u and v.

There are several ways to modify the predicted rating for a user according to the personality and trust factors. These strategies will be depicted in Section 4.2. Next, we will explain the aggregation functions that can be applied to combine the individual estimations.

4.1 Aggregation Functions

A wide set of aggregation functions has been devised for combining individual preferences [9], being the average and least misery strategies the most commonly used. In the experiments presented in this paper we use the average satisfaction strategy, ir 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 representing this strategy is:

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

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.

4.2 Modifying Individual Predictions with Social Elements

Our recommendation approaches [5] consist on evaluating the different behaviours that people have when reaching a decision making process. To do so we modify the predictions made by the individual recommender with the personality and trust factors. In that way not all the predictions are taken into account equally. We use two different methods to compute the new individual rating ($pred'(i,u)$) used in Equation 2.

- **Delegation-based method:** The idea behind this method is that users create their opinions based on the opinions of their friends. The estimation of the delegation-based rating ($dbr(u,i)$) given an user u and an item i is computed in this way:

$$pred'(u,i) = dbr(u,i) = \frac{1}{|\sum_{v \in G} t_{u,v}|} \sum_{v \in G \wedge v \neq u} t_{u,v} \cdot (\, pred(v,i) + p_v \,) \tag{4}$$

  In this formula, we take into account the recommendation $pred_{v,i}$ of every friend v for item i. This rating is increased or decreased depending on her personality (p_v), and finally it is weighted according to the level of trust ($t_{u,v}$). Note that this formula is not normalized by the group size and uses the accumulated personality. Therefore, this formula could return a value out of the ratings range. This is simply managed by the recommender by choosing the closest value within the valid range.
- **Influence-based method:** This method simulates the influence that each friend has in a given person. Instead of creating a new preference, it supposes that the user may modify her preference for an item depending on the preferences given by her friends to the same item, as shown in the following formula:

$$pred'(u,i) = ibr(u,i) = pred(u,i) + (1 - p_u)\frac{\sum_{v \in G \wedge v \neq u} t_{u,v} \cdot (\, pred(v,i) - pred(u,i)\,)}{|G| - 1} \quad (5)$$

In this formula, the individual rating for the item ($pred_{u,i}$) is modified according to its difference with the ratings of other users ($pred_{v,i} - pred_{u,i}$). This difference takes into account the trust between users ($t_{u,v}$). Finally, the accumulated difference is weighted according to our personality in an inverse way $(1 - p_u)$.

Next section presents the main contribution of this paper, a new group recommendation strategy, that uses the information retrieved by the ARISE architecture, personality, trust and personal preferences in order to provide a group recommendation based on alliances. It consists on a new approach to modify individual predictions with social elements, different from the *delegation-based* and *influence-based* methods that we have just explained.

5 Alliance Based Approach

Alliance based approach first computes personality and trust for every user in the group as explained in section 3. Next step uses this information to identify the leader users and her close friends set. Every user with a personality higher than a threshold α is considered a group leader. In Section 6 we use α as the 85% of the highest personality value in the group. Note that the number of group leaders is not fixed. We have empirically discovered in our case of study that our method performs better when we obtain a number of leaders close to half of the size of the group. For every leader in the group l, we obtain her close friends set $cfs(l)$. This set is obtained using the trust values computed between the leader and every other user in the group and then selecting the users that the leader trusts higher. This set represents all the "possible alliance mates". If the trust between a user, u_i and the leader l is higher than another threshold, β, she is included in her $cfs(l)$.

Negotiation between l and $cfs(l)$ begins to agree on a common product that the leader l likes. This negotiation process allows us to determine whether the proposal made by the leader is accepted or not. It runs as follows:

1. For every user in the group we obtain the individual estimation of ratings of the products in the catalogue. We use the *Individual preferences estimation* module of the ARISE architecture (see Section 3) by applying an individual recommendation approach with the information retrieved in the *explicit individual preferences* module. The construction of the recommender runs as follows.
2. Analyze the recommendations made to the leaders and identify which are their favourite items. This set of items, $lfi(l)$ (leaders favourite items), are the ones that each leader proposes to her close friends set $cfs(l)$ in order to

form the alliance. Note that the size of $lfi(l)$ is not fixed, it can be adjusted depending on the size of the catalogue of items. There are n ($n = |l|$) sets of leaders favourite items ($lfi(l)$), one for each leader in the group.
3. Propose the leaders individual favorite items $lfi(l)$ to leader l "possible alliance mates". A proposal is accepted if the estimated rating that a user u_i, with $u_i \epsilon cfs(l)$, has of the proposed item p_i, with $p_i \epsilon lfi(l)$, is higher than a certain threshold δ. This threshold δ is modified depending on the users personality (it will be bigger with stronger personalities) and also depending on the trust with the leader (if the user has a strong trust on the leader the threshold will be lower). See Equation 6 in Section 6.
4. When an user accepts the proposal we include her in the alliance of that leader. We note that the leader has θ ($\theta = |lfi(l)|$) attempts to "persuade" each one of the users in her $cfs(l)$, one attempt for every item in the set of the leaders favourite items. To be part of the alliance a user just has to accept one item of the proposed list. As we have said before, a leader l creates alliances $alli(l, p)$ with other users supporting a concrete product p_i. If the size of the alliance $|alli(l, p)|$ is greater than a half of the group, the items in $lfi(l)$ are directly chosen as the items for the group. If there is no majority we will choose the items proposed by the larger coalition.

6 Case Study: Movie Recommendation

In this section we evaluate the alliance based approach for group recommendation using the movie recommendation domain. The goal of the experiment is improving other group recommender approaches. We compare the results obtained using alliances with a base group recommender system using the average satisfaction aggregation function and also with our previous approaches using personality and trust [4,5]. The construction of the alliances recommender involves the processing of several factors that are obtained in different ways. The personality values are obtained through the TKI tests [11], whereas trust values are directly extracted from a social network where all the users belong to. Next we explain how we extract the information required from our users, how we measure the results, the configuration of our alliances recommender and the results of the experiment.

6.1 Experimental Setup

In order to perform our experiment in the movie recommendation domain, we created two events in two different social networks, Facebook[2] and Tuenti[3]. In these events we asked some of our users to complete three questionnaires[4]. The first questionnaire serves to obtain the personality of each user, is the one run by the *personality module*. Second questionnaire gets the individual preferences

[2] `http://www.facebook.com`
[3] `http://www.tuenti.com`. The most popular social network in Spain.
[4] Questionnaires are accessible at `http://www.lara.warhalla.com/` (in Spanish.)

of the user about cinema. Users have to evaluate 50 heterogeneous movies from the MovieLens data set [28](rating them with a range of 0.0 to 5.0). These 50 movies are the list of products that are assigned to each agent, and they are stored in the *Explicit individual preferences module*.

Finally, third test asks users to choose their 3 favourite movies from a list of 15 *recent* movies (of the 2009 year), that represents a movie listing from a cinema. This list of 15 products is the one gathered by the *Product Data module*. The movie listing was chosen from movies of the MovieLens database using a diversity function. The 3 movies selected by each user are included as her individual favourites, *if*. These movies are the ones she would actually like to watch or had enjoyed best. The answers to these questionnaires are analysed to define the user profile of each participant. 58 real users have participated in our experiment.

To measure the accuracy of the group recommendation we brought our users together in person and ask them to mix differently several times and simulate that they are going to the cinema together, forming different groups that would actually come out in reality. We provide them the 15 movies that represent our movie listing and we ask them to choose in the group which 3 movies in order they actually would watch together. We manage to gather 10 groups: 6 groups of 5 members and 4 groups of 9 members. The three movies that each group chooses are stored as the *real group favourites* set –*rgf*–. This way, to evaluate the accuracy of our recommender we can compare the set proposed by the recommender –the *pgf* set– with the real preferences *rgf*. The evaluation metrics applied to compare both sets are explained in Section 6.2.

Our group recommendation strategies combine individual recommendations to find an item (movie) suitable for any user in the group. This individual recommender is built using the jCOLIBRI framework [29] and follows a content based approach [30] to find the most similar movie rated by the user. It uses product descriptions and returns the collection of products that are more similar to the aimed product, assigning the rating given by the user as a prediction. This set of movies is different for each user and it has the information retrieved from the second questionnaire.

6.2 Evaluation Metrics

Our experiment requires an evaluation function to measure the accuracy of the group recommendation. To do so, we compare the results of our recommender system to the real preferences of the users (that is, what would happen in a real life situation). When we started our evaluation process we took into account the number of estimated movies that we were going to take into account. We are not interested on a long list of ordered items that estimates movies a user or group should watch. Real users are only interested on a few movies they really want to watch. This fact discards several evaluation metrics that compare the ordering of the items in the real list of favourite movies and the estimated one (MAE, nDCGs, etc.). On the other hand, the number of relevant and retrieved items in our system is fixed. Therefore, we cannot use general measures like recall or precision. However, there are some metrics used in the Information Extraction field [31] that limit the retrieved set. This is the case of the *precision@n* measure that computes the *precision* after n items have been retrieved. In our case, we can use the *precision@3* to evaluate how many of the movies in pgf are in the rgf set (note that $|rgf| = 3$). This kind of evaluation can be seen from a different point of view: we are usually interested on having at least one of the movies from *pgf* in the *rgf* set. This measure is called *success@n* and returns 1 if there is at least one hit in the first n positions. Therefore, we could use *success*@3 to evaluate our system computing the rate of recommendations where we have at least one-hit in the real group favourites list. For example, a 90% of accuracy using *success*@3 represents that the recommender suggests at least one correct movie for the 90% of the evaluated groups. In fact, *success*@3 is equivalent to having *precision*@3 $> 1/3$. We can also define a 2*success*@3 metric (equivalent to *precision*@3 $> 2/3$) that represents how many times the estimated favourites list *pgf* contains at least two movies from *rgf*. Obviously, it is much more difficult to achieve high results using this second measure.

6.3 Alliance Recommender System

For each group we build the alliance recommender using the following steps:

1. We obtain the members of the group and we calculate an estimation of their individual preferences with content based individual recommender system. After this process what we have is an estimated rating of each user for each of the 15 movies in the movie listing from the cinema.
2. We identify the leaders of the group, which are those who have a personality that is higher than the 85% of the personality value of the user with the strongest personality in the group (threshold α).
3. For each of the leaders we try to find alliances. To find the possible candidates that could form the alliance we select those users who have a trust with the leader higher than the 75% of the trust value of the most trusted user of the leader (threshold β).
4. To accept a user as part of the coalition, we propose the 3 movies that the leader of the group has with the higher rating, that as we remember we obtained from the individual recommender. If the users predicted rating for that movies is higher than threshold δ then the user is accepted as part of the alliance. Threshold δ is obtained with the following formula:

$$\delta = ir_{u,5} - t_i + p_r \tag{6}$$

where $ir_{u,5}$ is the predicted rating of the best fifth item for the user, $t_i = \mu * trust_{u,leader}$, and $p_r = \lambda * p_u$. $trust_{u,leader}$ represents the existing trust between the user and the leader, p_u is the personality value of the user and μ and λ have been experimentally obtained. ($\mu > 0.4$ and $\lambda < 0.5$).

We have built another alliance recommender system simplifying this last formula, we call it *Alliance-based Recommender simpler version*, we have done this to study the influence and benefits of using the trust and personality factors in order to vary the threshold of acceptance of a proposed item δ. This variation of our method obtains the threshold δ with this simplified formula:

$$\delta = ir_{u,5} \tag{7}$$

where $ir_{u,5}$ is the predicted rating of the best fifth item for the user.

5. After forming all the alliances we compare the sizes of the alliances. If the size of the alliance is greater than a half of the group we propose as selected items, the favourites of the leader, which are the 3 movies that the leader of the group has with the higher rating.

6.4 Experimental Results

In Figure 2 we have analyzed the performance of the base recommender, a group recommender using the same data-set but applying our influence-based recommendation method, a group recommender using the same data-set applying our delegation-based recommendation method, a group recommender with the simplified version of our alliances approach (the one that does not use personality and trust factors in order to calculate the threshold of acceptance of each item) and finally a recommender with our alliances approach. We can see that we have improved the performance of the basic recommender in a 10% with the success@3 and in a 40% with the 2success@3. Results also show that with the 2success@3 measure the alliances approach obtains the best results. As we have explained before this measure is much more difficult to obtain than the success@3 measure, so with this results we validate our alliances method and conclude that

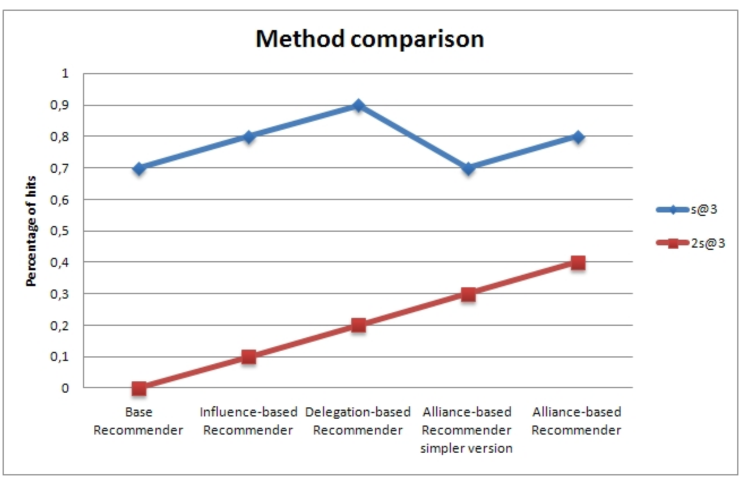

Fig. 2. Comparison of the results obtained with the base recommender, the influence-based recommender, the delegation-based recommender, the alliance-based recommender simpler version and alliance-based recommender

with it we improve our previous group recommendation strategies. From this Figure we also observe that it is essential to include the personality and trust factors in order to calculate the threshold of acceptance of each item, because with the simplified version of our alliance approach the results with the success@3 measure are equal to the base recommender so we do not improve with it the group recommendation. We must note that we still can validate our strategy because for the 2success@3 even with the simplified version of our alliance approach results are better than the ones obtained by the base, influence-based, and delegation-based recommenders.

7 Conclusions

In this paper we have proposed and evaluated a group recommendation strategy based on alliances for the recommendation of products in social networks. In previous papers we have already experimented with a novel method of making recommendations for groups taking into account the group personality composition and the social structure of the group. Once shown that personality profiles can improve a recommendation for a group of people, we have extended this approach by reflecting in a more realistic way the social relationships between the users involved in the recommendation. We have tested our method in the movie recommendation domain and shown that group recommendation using alliances improves the base group recommender system using the average satisfaction aggregation function. Results also have shown that with the 2success@3 measure the alliances approach obtains the best results and improve our previous group recommendation strategies. We have also observed that it is essential to include the personality and trust factors in order to calculate the threshold of acceptance of each item in the recommender system. Our proposed alliance based approach for group recommendation is based on identifying users with strong personalities try to create *alliances* with other users to support their personal preferences. This way, influencer users obtain the required votes to get their proposal chosen by the group. These influencers, or leaders, try to influence other users and they use their leadership to create the alliance. The proposed method first computes personality and trust for very user in the group and then uses this information to identify the leader users and her close friends set. Negotiation between the leader and people from her close friends set begins to agree on a common product that the leader likes. This negotiation process allows us to determine whether the proposal made by the leader is accepted or not. Our ongoing work consists on making further evaluations of our alliances method by embedding it into a social network application, where we will be able to continue our experiments with larger and more general populations.

References

1. Bridge, D., Göker, M.H., McGinty, L., Smyth, B.: Case-based recommender systems. Knowledge Engineering Review 20, 315–320 (2006)

2. Jameson, A., Smyth, B.: Recommendation to groups. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 596–627. Springer, Heidelberg (2007)
3. 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, New York (2009)
4. Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B.: Personality and social trust in group recommendations. In: Procs. of the 22th Int. Conference on Tools with Artificial Intelligence, ICTAI 2010, pp. 121–126. IEEE Computer Society, Los Alamitos (2010)
5. Quijano-Sánchez, L., Recio-García, J., Díaz-Agudo, B., Jiménez-Díaz, G.: Social factors in group recommender systems. In: ACM-TIST, TIST-2011-01-0013 (in press, 2011)
6. Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B., Jiménez-Díaz, G.: Happy movie: A group recommender application in facebook. In: 24th International Florida Artificial Intelligence Research Society Conference, FLAIRS 2011 (2011)
7. 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)
8. 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)
9. 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)
10. Chen, Y.L., Cheng, L.C., Chuang, C.N.: A group recommendation system with consideration of interactions among group members. Expert Syst. Appl. 34, 2082–2090 (2008)
11. Thomas, K., Kilmann, R.: Thomas-Kilmann Conflict Mode Instrument, Tuxedo, N.Y (1974)
12. Josang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision, pp. 618–644 (2007)
13. 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)
14. Miller Mcpherson, L.S.L., Cook, J.M.: Birds of a feather: Homophily in social networks, pp. 415–444 (2001)
15. Lazarsfeld, P., Merton, R.: Friendship as a social process: A substantive and methodological analysis. In: Berger, M., Abel, T., Page, C. (eds.) Freedom and Control in Modern Society, pp. 18–66. Van Nostrand, New York (1954)
16. Gladwell, M.: The tipping point: How little things canmake a big difference. Little Brown, Boston (2000)
17. Burt, R.S.: Toward a structural theory of action: Network models of social structure, perception and action, pp. 1336–1338 (1982)
18. McPherson, M., Smith-Lovin, L.: Homophily in voluntary organizations: Status distance and the composition of face-to-face groups, pp. 370–379 (1987)
19. Chalkiadakis, G., Elkind, E., Markakis, E., Polukarov, M., Jennings, N.R.: Cooperative games with overlapping coalitions. J. Artif. Intell. Res. (JAIR) 39, 179–216 (2010)

20. Barton, L., Allan, V.H.: Methods for coalition formation in adaptation-based social networks. In: Klusch, M., Hindriks, K.V., Papazoglou, M.P., Sterling, L. (eds.) CIA 2007. LNCS (LNAI), vol. 4676, pp. 285–297. Springer, Heidelberg (2007)
21. Airiau, S., Sen, S.: On the stability of an optimal coalition structure. In: Proceedings of 19th European Conference on Artificial Intelligence, ECAI 2010, Lisbon, Portugal, August 16-20, pp. 203–208 (2010)
22. Nuno, D., Ao, S.J.S., Helder, C.: Agent-based social simulation with coalitions in social reasoning. In: Moss, S., Davidsson, P. (eds.) MABS 2000. LNCS (LNAI), vol. 1979, pp. 244–265. Springer, Heidelberg (2001)
23. Elkind, E., Chalkiadakis, G., Jennings, N.R.: Coalition structures in weighted voting games. In: 18th European Conference on Artificial Intelligence, ECAI 2008. Frontiers in Artificial Intelligence and Applications, vol. 178, pp. 393–397. IOS Press, Amsterdam (2008)
24. McGinty, L., Smyth, B.: Collaborative case-based reasoning: Applications in personalised route planning. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, pp. 362–376. Springer, Heidelberg (2001)
25. Ontañón, S., Plaza, E.: An argumentation-based framework for deliberation in multi-agent systems. In: Rahwan, I., Parsons, S., Reed, C. (eds.) Argumentation in Multi-Agent Systems. LNCS (LNAI), vol. 4946, pp. 178–196. Springer, Heidelberg (2008)
26. Sinha, R.R., Swearingen, K.: Comparing recommendations made by online systems and friends. In: DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries (2001)
27. Recio-García, J.A., Díaz-Agudo, B., González-Calero, P.A.: Prototyping Recommender Systems in jCOLIBRI. In: Proceedings of the 2008 ACM Conference on Recommender Systems, pp. 243–250. ACM, NY (2008)
28. Bobadilla, J., Serradilla, F., Hernando, A.: Collaborative filtering adapted to recommender systems of e-learning. Knowl.-Based Syst. 22, 261–265 (2009)
29. 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)
30. 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)
31. Oard, D.W., Baron, J.R., Hedin, B., Lewis, D.D., Tomlinson, S.: Evaluation of information retrieval for e-discovery. Artif. Intell. Law 18, 347–386 (2010)