Development of a Group Recommender Application in a Social Network

Lara Quijano-Sánchez[a], Belén Díaz-Agudo[a], Juan A. Recio-García[a]

[a]*Department of Software Engineering and Artificial Intelligence, Universidad Complutense de Madrid, Spain*

Abstract

In today's society, recommendations are becoming increasingly important. With the advent of the Social Web and the growing popularity of Social Networks, where users explicitly provide personal information and interact with others and the system, it is becoming clear that the key for the success of recommendations is to develop new strategies which focus on social recommendations leveraged by these new sources of knowledge. In our work, we focus on group recommender systems. These systems traditionally suffer from a number of shortcomings that hamper their effectiveness. In this paper we continue our research, that focuses on improving the overall quality of group recommendations through the addition of social knowledge to existing recommendation strategies. To do so, we use the information stored in Social Networks to elicit social factors following two approaches: the cognitive modeling approach, that studies how people's way of thinking predisposes their actions; and the social approach, that studies how people's relationships predispose their actions. We show the value of using models of social cognition extracted from Social Networks in group recommender systems through the instantiation of our model into a real-life Facebook movie recommender application.

Keywords: Social Networks, Social Knowledge, Group Recommender Systems, Social Applications

1. Introduction

Recommender systems are knowledge-based systems which support human decision-making. In an era of overwhelming choice, they help us decide which products, services and information to consume. The focus of attention in recommender systems' research and development has been on making recommendations to individual consumers (e.g. see [1] for an overview). These places focus on the easier case, but ignore the fact that it is as common, if not more common, for us to consume items in groups such as couples, families and parties of friends. The choice of a date movie, a family holiday destination, or a restaurant for a celebration meal all require the balancing of the preferences of multiple consumers.

This kind of recommendations are provided by group recommender systems [2]. These systems commonly aggregate real or predicted ratings for group members [2, 3, 4, 5]. The aggregation functions typically used are inspired by the social welfare functions developed by the Social Choice Theory research [6]. First, for each group member, an individual recommender system predicts a set of ratings for the candidate items. Next, the group recommender aggregates the ratings: for each candidate item, it might take the average of group members' ratings, or the minimum, or the maximum, for example. Finally, it recommends to the group the items with the highest aggregated ratings.

However, this widely accepted approach for group recommendations ignores the social factors that influence real group decision-making. In real-life group decision-making a variety of social-cognitive processes underlie the choices that people make [7]. For example, it has been proven that social media highly influences peoples' decisions, relationships, and education. Several researchers study the impact of social media in our lives [8]. These studies evaluate the social context, which 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. Besides, circumstantial life events, influences, and surroundings can further change our social abilities [9]. In our research, we study how to model these social-cognitive

Email addresses: lara.quijano@fdi.ucm.es (Lara Quijano-Sánchez), belend@sip.ucm.es (Belén Díaz-Agudo), jareciog@fdi.ucm.es (Juan A. Recio-García)

processes in human decision-making processes [10, 11]. Specifically, we have designed a method that is aware of the different personalities and social ties that group members present. These techniques and their associated algorithms have been compiled into a generic architecture named ARISE (Architecture for Recommendations Including Social Elements) that can be instantiated into any kind of social recommender system[1]. The common and key factor in all the different types of recommenders that can be built in all sort of domains using this generic architecture is the inclusion of social elements. These social elements define each person 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 [8]. Besides, we have provided a software development methodology to ease the instantiation of the ARISE architecture into concrete applications. This methodology is based on templates [12] that formalize the functional behaviour of social recommender systems and facilitate its configuration and deployment.

To illustrate and validate the capabilities of our *social recommendation* approach we have instantiated our model into a real-life recommender system: *HappyMovie*[2], which is a particular application of our generic architecture ARISE for the movies recommendation domain in the social network Facebook. *HappyMovie* serves us as a use case and experimental environment where we are able to evaluate our ARISE architecture and our *Social Group Recommendation Method* with real data.

The main contribution of this paper is to measure the benefits of having a social group recommender application embedded in one of the biggest Social Networks in the world[3] following two goals:

1) Testing in terms of users' acceptance, the benefits of using our *Social Recommendation* approach against a state-of-the-art approach that does not use social knowledge[4]. We provide a usability evaluation of the application *HappyMovie*. To do so, we have performed a functionality test where 60 users have evaluated different aspects of *HappyMovie*, proving that our application and its associated *Social Group Recommendation Method* indeed offer good and eligible group recommendations. With this evaluation we also want to measure users' response towards having the application in Facebook, if there is any resentment towards the different knowledge elicitation processes, and more importantly their satisfaction levels with the recommendations provided. This is an important evaluation as we want to test if *HappyMovie* is a viable and suitable platform to continue our research with.

2) Testing if users will indeed use the application and therefore it is worth making it public. This evaluation will allow us to determine if *HappyMovie* is viable not only for giving good recommendations but also as a research tool that allows us to extract group related knowledge[5].

Another contribution toward group recommender applications is a detailed explanation of our *Social Group Recommendation Method* in a social application like *HappyMovie*. This survey provides a self-contained full description to support the complete reproducibly of our system. In [13] we presented a functional vision of *HappyMovie*. This previous work introduced an initial version of *HappyMovie* where the focus was on presenting an experiment on the viability of making the application more easy going through an interactive game that measures users' personality instead of the long questionnaire that we previously used in [10, 11]. In this paper we have refined, tested and justified the ideas and decisions made in our previous work. With it, we want to provide future group recommender developers a reference on how to build and exploit further *social group recommender* systems. The novelty presented in the paper lies in the explanation of how we now elicit and apply each social factor in the recommendation process and in the explanation of why these factors improve the overall quality of the recommendation.

The remainder of this paper is structured as follows: In the next section we introduce some of the state-of-the-art knowledge regarding recommender systems and study other recommendation approaches whose main

[1]Note that the verification of the generalisability of our approach is out of the scope of this paper. However, it has already been proven valid in two different domains: movies & clothing, as stated in [12].

[2]http://gaia.fdi.ucm.es/research/happymovie

[3]Facebook passes 1.19 billion monthly active users.

[4]Note that in [10, 11] we proved in simulated environments that using social factors, i.e. personality and trust, improved the recommendation accuracy. However, now that we have embedded our approach directly in a social network, the elicitation of these factors has been adapted to this situation and is therefore different. Hence, new experiments need to be carried out.

[5]As a future goal we would like to obtain a big dataset with all the knowledge that this application is capable of acquiring, in terms of trust levels, users' personality, users' preferences and final group decisions. With this dataset we could be able to provide to the recommenders community a public group dataset (we are not aware of any similar public dataset), run Big Data experiments or study group similarity composition and behaviour among others.

goal is also the improvement in the performance of recommenders through the usage of social information. Next, Section 3 introduces the details of our Facebook application *HappyMovie*. The evaluation of *HappyMovie* is presented in Section 4. Finally Section 5 concludes the paper.

2. Literature Review

There are a lot of domains where group recommendation techniques can be applied, for example in the music domain, MusicFX [14] and FlyTrap [15]. Polylens [16] in movies domain. Or, regarding recommendations of restaurants for groups, Pocket Restaurant Finder [17]. 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 user behaves in a group when joining a decision-making process. Some works, do take into account not only the preferences of every member but also the interaction among them; Travel Forum Decision [18] or the work performed by Masthoff and Gatt [6] are examples of this.

Social Networks have been one of the most important topics in the last few years, with nets like Facebook, Twitter and MySpace, among others. The use of Social Networks and trust when building recommender systems is not new [19]. 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 [20]. There is a huge body of work about the generation of trust models. There are specific approaches that use a custom trust network to recommend items. One example is FilmTrust [21], which exploits a custom network of trust among users according to movie preferences. However, these specific trust networks are quite difficult to generate because they require explicit feedback from users, and these can generate rejection.

The general opinion is that there is a need to adapt the recommendation process to group composition [2, 22]. 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 [3].

Avoiding repeated recommendations is also an important matter that some works give special importance to. For example [22] in the recommendation of TV programs domain or FlyTrap [15] where previous selections are also taken into consideration. Besides, some recommendations tend to be repeatedly detrimental to the same group members. Another system that takes previous selections into account is PoolCasting [23]. In our approach, we have given special treatment to the evaluation of previous recommendations in order to avoid this tendency and, similarly to Masthoff & Gatt's work [6], we have gone one step forward considering also users' satisfaction with past recommendations.

Regarding the extraction of information from Social Networks, in the last few years there has been a huge line of research that uses social information to improve recommender systems. Examples of this are [24] that use the social information stored in the music Social Network *last.fm*, and capture explicitly expressed bonds of friendship as well as social tags to improve recommendations' accuracy. Or [25], that use social tags to recommend the most suitable multimedia contents for users.

While recommender systems have been extensively researched since the mid-1990s [14, 26], the study of social-based recommender systems is a new area [27]. One of the key factors that social-based recommendations use is the study of the multiple dimensions within a user's Social Network, including social relationship strength, interests, and user similarity. In [28] the authors seek to develop novel group recommender systems that leverage these dimensions.

Users in today's online Social Networks often post a profile, consisting of attributes such as geographic location, interests, etc. Such profile information is used on recommender systems as a basis for grouping users, for sharing content, and for suggesting users who may benefit from interaction. However, in practice, not all users provide these attributes and several researchers have focused on handling imprecise and incomplete information [29]. In [30] they use the information stored in online Social Networks to infer the attributes missing in some users' profiles.

As we can see, social relations provide an independent source for recommendation; various approaches are proposed to build social recommender systems such as trust relationships [31], trust propagation [32], or directly trust user based recommenders [33]. Besides, there is recent work reporting significant recommendation performance improvement for social recommender systems [21, 34, 35, 36, 37, 38, 39]. On the other hand, there are also unsuccessful attempts at applying social recommendation [40, 41].

Summing up, social recommendation is still in the early stages of development, and there are many challenging issues needing further investigation. Following this reasoning we consider the necessity to discuss and propose new research directions that can improve social recommendation capabilities and make social recommendation applicable to an even broader range of applications. Besides, we have not found any work (aside from our own) that integrates an automatic elicitation of the knowledge stored in users' profiles to obtain a trust value that is later on used along with a personality value and a memory of past recommendations to improve the results of group recommender systems.

Next, we introduce our *Social Group Recommendation Method* and how we have modelled it in our *Social Group Recommender* application *HappyMovie*.

3. *HappyMovie*

Our group recommendation method is based on three major components: personality, social trust and memory of past recommendations. The integration of these social factors enables a realistic simulation of decision-making processes followed by groups of people when deciding a joint activity. With *HappyMovie* we try to mitigate certain limitations in existing group recommender systems, like obtaining users profiles or offering trading schemes in order to reach a final agreement [42]. It serves us as a use case and experimental environment where we can evaluate our *Social Group Recommendation Method* with real products. This way we can validate our previous results [10, 11], where we concluded that personality and social trust factors indeed improved the performance of group recommendations. In addition, thanks to the inclusion of the system in a social network we can now continue our investigation with further experiments that include the study of users' response over time towards the recommendations given, a detailed analysis of their opinion towards *HappyMovie*'s recommendations and obtaining a large set of data which will enable us to better study group similarities and recurring behaviour (this last experiment is out of the scope of this paper but would be now possible thanks to the social factors' automatic elicitation process now granted thanks to having the system embedded in a social network).

Our goal with *HappyMovie* is to evolve and integrate group recommender systems into the Social Web –concretely Facebook– where personal relations can be analyzed and exploited to enhance the process of making recommendations to groups. Within this environment, we are able to infer much of the information needed to perform *Social Group Recommendations* directly from Social Networks [21, 43]. Previously, the acquisition of such social data had to be performed by means of tedious questionnaires. The integration into

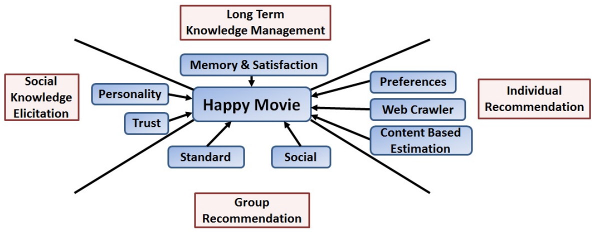

Figure 1: *HappyMovie*'s modules

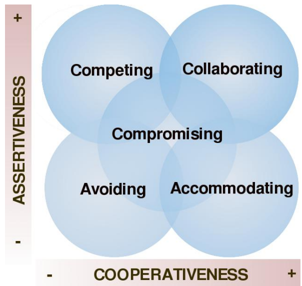

Figure 2: TKI personality modes

Social Networks eases this process and provides a lot of valuable feedback to evaluate and improve our proposal. In our social method, we have reproduced people's behaviour based on the idea that the relationship between individuals and their networks of people directly influence their lives [8].

The architecture of *HappyMovie* is represented in Figure 1. The application is organized in four different groups of modules: Social Knowledge Elicitation, Long Term Knowledge Management, Individual Recommendation and Group Recommendation. The next sections detail and analyse each of them.

3.1. Social Knowledge Elicitation Modules

When people face situations where their interests or preferences are incompatible with others *conflict situations* arise. Here conflict is understood as a difference that prevents agreement. More concretely, in group interactions it is defined as a competitive or opposing action of incompatibles: antagonistic state or action (as of divergent ideas, interests, or persons) [44]. Different people have different expectations and behavior in conflict situations [6]. Our research to improve group recommendation systems studies the different behaviors that people have in conflict situations according to their personality and inter-personal trust. Next we describe the modules in *HappyMovie*'s architecture in charge of eliciting users' social knowledge.

Personality modelling Modelling human cognitive reactions through computer interfaces is not new

[45, 46]. As our goal is to estimate users' behavior in decision-making processes (users' personality value), *HappyMovie*'s users perform an adaptation of the Thomas-Kilmann Conflict Mode Instrument (TKI) test [47], which is a leading instrument used by individuals and businesses for identifying their ability to handle conflicts in decision-making processes. This test is commonly used in the human-machine interaction area, due to its efficiency, easy evaluation and easy usage for people not related to the psychology area. In comparison with other similar tests, such as *Ego Gram* [48], that measures personality according to three "Ego States" (Parent, Child, Adult), or *Pen Model* [49], that measures Psychoticism, Extraversion and Neuroticism (PEN), we believe that the TKI test is the most suitable choice as its main focus is on measuring people's reactions in conflict situations whereas other tests' focuses (even though they may perform a wider personality study) are not as specific for our purpose as TKI is. Besides, TKI provides a tangible and measurable value, easy to interpret and most importantly short and easy to answer. We think that this last characteristic is a key element in the success and acceptance of *HappyMovie* as users may not be willing to answer long personality tests in order to obtain a movie recommendation. For example, the NEO-PI-R [50] is a 240-item questionnaire (designed to operationalize the five-factor model of personality [51]), which for our recommendation purposes is too long. The TKI test consists of 30 different situations with two possible answers. Depending on the answers, a score is assigned for 5 existing personality modes (see Figure 2) organized according to two dimensions: *assertiveness* and *cooperativeness*.

Once we finished our experiments in simulated environments [10, 11], users were asked to give their opinions about the test which they described as tedious and long. To make the application more easy going we studied the possibility of using a movie metaphor as an alternative personality test. Consequently we developed an alternative metaphor that lightens this activity. This interactive metaphor consists of displaying two movies characters with opposite personalities for each of the 5 existing conflict-handling modes. One character represents the essential characteristics of the mode, while the other one represents all the opposite ones. Users have to move an arrow showing their degree of similarity with the personality characteristics of the characters being presented (these personality characteristics along with examples of typical behaviour patterns are presented under each character's image). In [13] we concluded that it is possible to replace the TKI personality test with the movie metaphor test because it provides an statistically confirmed accurate estimation of the personality mode. The results that the TKI test provided for the five different personality modes in comparison with the values that the movie metaphor test gathered had a Mean Absolute Error (MAE) [52] of 0.12. Hence, we believe that it is worth sacrificing a little accuracy in the test results (as they are not for psychology testing purposes) in exchange of enhancing significantly the usability and interest for the application. A screenshot of *HappyMovie*'s personality test is presented in Figure 3.

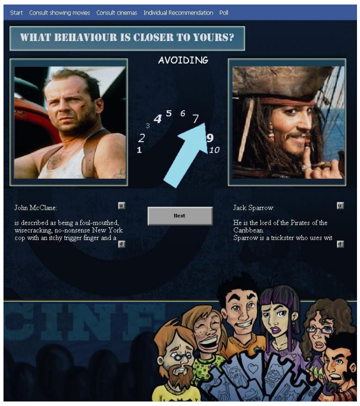

Figure 3: *HappyMovie*'s personality test

In this paper, we fully explain how to calculate the personality value p_u, which 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.

Firstly, users select the character they feel more identified with by moving the arrow. This process is repeated for the 5 personality modes with different representative and opposite characters. The arrow has positions (marked 1 to 10) that represent the percentage that users have in each category. This is done to measure users' resemblance degree on each personality mode.

The next step is to reduce these 5 different types of personality into TKI's two dimensions: *assertiveness* and *cooperativeness*. If one of the 5 scores, that have been now estimated, is below or above the 25 or 75 percentile according to the population, then users are classified as having a low or high personality mode in that category [53]. This way the test indicates if users have high or low degree of resemblance with each one of the existing modes. Following the schema shown in Figure 2, if users have high competing and collaborative mode they are assigned a high assertiveness value. High avoiding and accommodating personality modes are considered as low assertiveness. Following the second dimension, high cooperativeness value is given to users if they have high accommodating and collaborat-

TKI Mode	Assertiveness		Cooperativeness	
	High	Low	High	Low
Competing	0.4	-0.2	-0.2	0
Collaborating	0.4	-0.1	0.2	-0.1
Accommodating	-0.2	0	0.4	-0.1
Avoiding	-0.4	0.2	-0.4	0.2
Compromising	0	0	0	0

Figure 4: Weights used to obtain the TKI modes

TKI Mode	Percentil	Classification
Competing	16.7%	Low
Collaborating	24.0%	Low
Accommodating	76.0%	High
Avoiding	83.3%	High
Compromising	58.3%	Not Relevant

Figure 5: Example of the calculation of the TKI modes

ing mode. The assertiveness and cooperativeness values are a weighted sum of the five personality modes. These weights are the coefficients shown in Figure 4[6]. For example, users with a high percentile score in Competing mode add a weight of 0.4 in Assertiveness and -0.2 in Cooperativeness. Medium percentile scores are not included in the personality estimation.

For the sake of clarity we here present an example of the calculation of these values: let's say that a user u has a percentage of the personality modes as shown in Figure 5, then, assertiveness and cooperativeness values are calculated as in Equations 1 and 2 respectively.

$$Assertiveness(u) = -0.2 - 0.1 - 0.2 - 0.4 + 0 = -0.9 \quad (1)$$

$$Cooperativeness(u) = 0 - 0.1 + 0.4 - 0.4 + 0 = -0.1 \quad (2)$$

Once the 5 personality modes are reduced to the *assertiveness* and *cooperativeness* dimensions, the personality value (p_u) is computed as the difference between both dimensions. P_u represents user u's predominant behavior according to her/his TKI evaluation, i.e., how assertive or cooperative s/he is. This is computed using the following equation:

$$p_u = \frac{1}{2}(1 + Assertiveness(u) - Cooperativeness(u)) \quad (3)$$

[6]Note that although personality computation is performed equally in [54, 11], weights are slightly different as they now use TKI's updated normative sample [53].

Next we explain how we compute the other social factor involved in our method, trust[7].

Trust Estimation

This module obtains the inter-personal trust or social tie between users. This factor can be estimated following different approaches, being most of them manual [19], task that users resented and found very tedious. Hence, we propose its elicitation from Social Networks. In this section we detail how the computation of the trust factor can be now automatically computed thanks to embedding the group recommender system in a social network application[8]. The process consists of calculating the inter-personal trust by analysing users' profiles and interactions in the social network. Users in Facebook post a huge amount of personal information that can be extracted to compute the trust with other users: likes and interests, personal preferences, pictures, games, etc.

The use of trust and other social knowledge obtained from Social Networks in the development of recommender systems is not new [21, 55]. We have reviewed several existing works [56, 19] that identify the factors to be analysed. In order to move from theory to practice it is important to note that these factors are not easy to quantify and are limited by the social network's API extraction power.

Previous works have reported that trust and tie strength are conceptually different but that there is a correlation between them [57]. [58] defines tie strength as a (probably linear) combination of four factors: the amount of time, the emotional intensity, the intimacy (mutual confiding), and the reciprocal services which characterize the tie. The literature reviewed identifies these four factors as some of the major dimensions of predictive variables. With these dimensions as a guide, [56] identified 74 Facebook variables as potential predictors of tie strength. They presented a diagram showing percentages that indicate the predictive power of their top seven tie strength dimensions[9] and also the top three predictive variables for each dimension. From the

[7]Note that this trust factor measures the tie strength in people's relationships.

[8]Contrarily, for the experiments carried out in [10, 11] the trust factor had to be differently computed and was done by friending on Facebook a small sample of users and extracting the information needed manually. Then we used 10 variables to estimate the trust factor, whereas as we will next see in here we only use 4. This change in the trust computation is due to the computed relevance of those factors (the weight in the trust computation of some of them was almost insignificant in [11]) and the extraction power that Facebook gives us (status was impossible to automatically extract reliably).

[9]Note that [56]'s top four dimensions are the ones that [58] used as definition of tie strength and therefore the ones that we have adopted,

predictive variables that [56] provided, we selected the ones that were more representative of each major dimension and which could also be extracted from users' Facebook profiles (as we have said before, we are limited to the elicitation power that Facebook grants us). The percentages that [56] presented for [58]'s four dimensions were:

1. **Intimacy** (32.8%), predictive variables: Appearances together in photo, Participants appearances in photo, Distance between hometowns, etc. In our estimation we use overlapping photo tags.
2. **Intensity** (19.7%), predictive variables: Wall words exchanged, Inbox messages exchanged, Participant-initiated wall posts, etc. In our estimation we use all sorts of messages exchanged.
3. **Duration** (16.5%), referred to when users first met. We did not have access to the date when two people became friends. Hence, we were forced to compute it as the structural variable, number of mutual friends, where the more friends they have in common the longer they have interacted as more different groups of friends overlap.
4. **Reciprocal Services** (21.7% [10]), top predictive variables: Educational difference, Occupational difference, Links exchanged by wall post, Applications in common, etc. In our estimation we use overlapping profile information.

Summing up, in *HappyMovie*, the trust factor that estimates tie strength between users u and v is computed as follows[11] (Note that we understand that $t_{u,v}$ is a directed measure and that $t_{u,v} \neq t_{v,u}$):

$$t_{u,v} = 0.361 \cdot Intimacy_{u,v} + 0.239 \cdot ReciprocalServices_{u,v} + 0.219 \cdot Intensity_{u,v} + 0.181 \cdot Duration_{u,v} \quad (4)$$

Where the predictive variables are computed as follows[12]:

$Intimacy_{u,v}$: Represents how much users interact outside the social network. To compute it we evaluate the percentage of pictures they appear together in the last year (denoted by $percentage_{tags}$). Note that we compute this percentage as a proportion of the pictures in u's profile that v is tagged on. This is done to contemplate the possibility where u has not posted a lot of pictures but v appears in most of them.

$$Intimacy_{u,v} = \begin{cases} 1.0, & \text{if } percentage_{tags} \text{ is} > 75\% \\ 0.7, & \text{if } percentage_{tags} \text{ is} > 50\% \\ 0.5, & \text{if } percentage_{tags} \text{ is} > 25\% \\ 0.3, & \text{if } percentage_{tags} \text{ is} > 10\% \\ 0.1, & \text{if } percentage_{tags} \text{ is} < 10\% \end{cases}$$

$Intensity_{u,v}$: Represents how much users interact inside the social network. To compute it we count the number of interactions in the last year (denoted by *interaction*). We understand that there has been interaction in the social network if users have exchanged messages of any kind (private messages, wall messages, etc).

$$Intensity_{u,v} = \begin{cases} 1.0, & \text{if } interaction > 3 \text{ days a week} \\ 0.7, & \text{if } interaction > 1 \text{ day a week} \\ 0.5, & \text{if } interaction > 1 \text{ day a month} \\ 0.3, & \text{if } interaction > 1 \text{ day each 3 months} \\ 0.1, & \text{if } interaction < 1 \text{ day each 3 months} \\ 0.0, & \text{otherwise} \end{cases}$$

$Duration_{u,v}$: Represents how long they have known each other. We compute it as a structural variable that measures the number of common friends. We understand that the more friends they have in common the longer they have known each other because more different circles of friends overlap.

$$Duration_{u,v} = \begin{cases} 1.0, & \text{if } >25 \text{ common friends} \\ 0.7, & \text{if } >15 \text{ common friends} \\ 0.5, & \text{if } >10 \text{ common friends} \\ 0.3, & \text{if } >5 \text{ common friends} \\ 0.1, & \text{if } <5 \text{ common friends} \end{cases}$$

$ReciprocalServices_{u,v}$: Represents how similar their profiles are, in terms of common interests (music, movies, etc), common schools, jobs, visited cities, etc. To compute it we evaluate the percentage of common posted information (denoted by $percentage_{info}$). Note that we compute this percentage as the proportion of information (interests, personal information, etc) in u's profile that also appears in v's profile. This is done to contemplate the possibility where u has not posted a lot of information but v has also posted most of them.

$$ReciprocalServices_{u,v} = \begin{cases} 1.0, & \text{if } percentage_{info} \text{ is} > 75\% \\ 0.7, & \text{if } percentage_{info} \text{ is} > 50\% \\ 0.5, & \text{if } percentage_{info} \text{ is} > 25\% \\ 0.3, & \text{if } percentage_{info} \text{ is} > 10\% \\ 0.1, & \text{if } percentage_{info} \text{ is} < 10\% \end{cases}$$

as the literature has not resolved this issue, let alone specified how many discrete tie strength levels exist.

[10]This dimension was not represented by [56]'s variables, from their variable definition we understood it as Profile Distance (13.8%) + Services (7.9%)

[11]The weights presented in this equation are a proportion of the percentages of the top 4 variables presented by [56] (as [56]'s top 4 only sum 90,7%).

[12]Note that the thresholds and specific values are different from the ones used in [10, 11], as we have experimentally obtained them after analyzing average situations in the users' profiles that took part in our previous experiments.

We have tested our trust estimation with a small group of users that indicated us their real trust factor, and obtained and estimated MAE error of 0.16. Hence, although we cannot conclude this section with a design prescription, we are comfortable enough in presenting it as a useful estimation of the tie strength between users[13].

The trust calculation is done every time a user joins an event with the rest of users also attending to it. These values are not stored, but repetitively calculated as Facebook profiles keep changing and so does trust between two friends. Note that $t_{u,v} \in (0, 1]$, 0 being the reflection of a not very trusted person and 1 the reflection of a highly trusted one.

3.2. Long Term Knowledge Management Modules

HappyMovie stores all the recommendations that have been made for every user and every group. This feature avoids repeating past recommendations and also ensures a certain degree of fairness in the long run. Frequently, a group will expect to use the application several times, thereby getting a bigger sample of recommendations. However, our *Social Group Recommendation Method* tends to always favour the same users (because they have stronger personalities or because they are closer friends with other members). Therefore, we could end up with a situation where we have some dissatisfied users because we take their opinions less into account for the group's sake. In order to avoid a situation of high deviation in the satisfaction levels of the group, we must take into account users' satisfaction regarding past recommendations. It would be desirable that future recommendations favour dissatisfied users so that all of them reach a proper level of satisfaction.

To address this issue, we propose the use of a memory of past recommendations. This way, if one member accepts a proposal that s/he is not interested in, next time her/his preferences will be prioritized in the recommendation process. This means that her/his opinion will have a higher weight next time. These weights will also be influenced by the different personalities of each group member. For example, a user who dislikes the movie (gives it a low rating) may nevertheless be satisfied with the recommendation, especially if s/he appreciates that it has been necessary to balance conflicting interests. Her/his satisfaction might be all the greater if s/he has a more accommodating (less selfish) personality type, or if the recommendation better matches the tastes of group members with whom s/he has stronger connections through contagion and conformity [6]. This behaviour is modelled by immediately compensating users who have been negatively affected and have strong personalities and bearing in mind that users with mild personalities might not mind giving in several times.

The satisfaction value s_u is the level of satisfaction of a user u. A user who is extremely happy with the recommendations will have this satisfaction value close to 1. However, the more dissatisfied with the recommendations s/he is, the more that this value will decrease, reaching down to 0 in the worst case. An important and interesting issue of this approach is the time scope of the memory of users' satisfaction. We can update the s_u value to reflect the satisfaction according to the last immediate group recommendation or take into account previous ones. Therefore, the satisfaction value for an execution t of the recommender may depend on the satisfaction of the user with the items recommended in t but also depends on her/his satisfaction with the previous recommendations $t-1, t-2, \ldots$. Hence, we manage two satisfaction values:

Instant satisfaction (is_u): reflects the immediate user's satisfaction with the last recommendation. This is, her/his conformance with the last item recommended to the group. We ask users to rate the items being recommended to the group in order to obtain the instant satisfaction value [14].

Global satisfaction: (s_u): measures the average satisfaction of the user through time. It is updated every time a recommendation is made:

$$s_u(t) = (1 - \delta) \cdot is_u(t) + \delta \cdot s_u(t - 1) \tag{5}$$

In this equation we use the $\delta \in [0..1]$ threshold to adjust the impact of the previous satisfaction when updating that value. Somehow, this threshold measures the degree of forgetfulness about past (in)satisfaction. For example, some people could easily remember that they were not taken into account for the last recommendation when facing a new decision making process to select a similar item. On the other hand, other users won't ever

[13] We are currently performing an experiment with *HappyMovie*'s users where, after obtaining users' real trust value through enquiries, we use a genetic algorithm to adjust the weights of each variable. Our goal is to achieve the minimum MAE when comparing the real trust value given by users to the estimation that *HappyMovie* provides. Unfortunately, these results are not available at the time of writing so we leave them for future work.

[14] Note that for the experiments detailed in Section 4.4 and performed to justify the necessity of including *long term recommendations*, we computed is_u as the average of the three ratings that users give as feedback once a recommendation is presented to the group (see Figure 8).

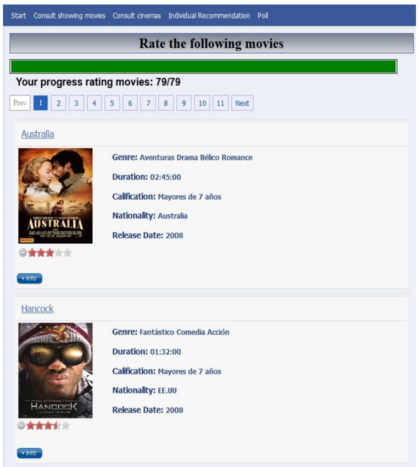

Figure 6: *HappyMovie*'s preferences test

take it into account. The measurement of this threshold belongs to the domain of the social sciences and is out of the scope of this paper. For the experiments presented in Section 4.4, we have configured a δ value of 0.5 to represent a balanced impact of previous satisfaction values.

3.3. *Individual Recommendation Modules*

Our group recommendation method is based on preference aggregation approaches [59, 60]. These approaches are based on the aggregation of users' individual ratings to obtain an estimated rating for the group. Hence, the basic building block of our group recommender is an individual recommender that computes the estimated preference of users for a given item. Individual recommendations in *HappyMovie* follow a content based approach [61]. This approach, schematized in Figure 1, uses the descriptions of the products to be recommended (obtained with the *Web Crawler module*), compares them with the descriptions of products rated by the user (obtained with the *Preferences Elicitation module*), and predicts the rating for the aimed products (computed in the *Content Based Estimation module*) by computing the average of the most similar rated products.

Preferences Elicitation

In the preferences elicitation test users indicate their taste in movies. The ratings here obtained are used by the individual recommender, that estimates the different movies to be recommended according to users' preferences in actors, genre, etc. For example, if a user has voted with 3 stars a certain movie, as we can see for example in Figure 6, we could consider that s/he likes that type of movies, so later, the individual recommender will analyse the characteristics of this movie and try to find a similar one. In order to complete the test, users must rate at least 40 movies through an 6-point Likert scale (0 to 5). Users are allowed to run this test on demand to modify or increase their ratings. The more ratings users give the more accurate their personal profile will be, and therefore the individual recommender will perform better. This test returns a set of real ratings $r_{u,i}$ for every user u in group G_a and item i in the test set T_s.

The test always presents the same 70 movies, which have been carefully chosen to cover a wide spectrum of movie tastes. They are the most popular movies of the last 3 years in all the different studied genres[15]. After testing different approaches Figure 6 shows how it has finally been implemented.

Web Crawler

We have built a Web Crawler that searches the web and retrieves all the movies and movie sessions being displayed in Spain's cinemas. This Web Crawler obtains a full technical datasheet for each of the movies being displayed. Each specific characteristic of the movie is a field that the individual recommender compares. For example, in our particular case study these characteristics are main actors, director and synopsis, between others. The retrieved set of movies, with all their specific information, is the target movie listing T_a containing the items i sent to the individual and group recommenders.

Content Based Estimation

We have chosen a content-based approach to estimate the rating users would assign to a new movie [61]. An alternative approach is a collaborative filtering approach [62]. However, we have chosen the first option because the movies to be recommended are the most recent movies on cinemas, so there are too new to have user ratings. Hence, we could not use those ratings as collaborative recommenders do. This section produces for every user u in the active group G_a a set $\{\hat{r}_{u,i} : u \epsilon G_a, i \epsilon T_a\}$ with the individual predicted ratings for all the target movies.

Our content-based method applies a weighted average of the similarity of the following fields that describe each movie: duration ($w_1 = 0.01$), recommended age ($w_2 = 0.03$), nationality ($w_3 = 0.11$), actors ($w_4 = 0.17$), directors ($w_5 = 0.178$), percentage of action

[15]The first time that we implemented this test [13], we chose well-known classic movies, however they were no good for the recommender as it compares actors and directors, besides genre, etc, from current movies on cinemas. So, if for example we selected *Marilyn Monroe* movies the actor's field would be useless as she is no longer making movies and there would not be any possible comparison between fields.

(w_6 = 0.042), percentage of animation (w_7 = 0.045), percentage of adventures (w_8 = 0.043), percentage of comedy (w_9 = 0.045), percentage of documental (w_{10} = 0.02), percentage of drama (w_{11} = 0.025), percentage of fantasy (w_{12} = 0.034), percentage of romantic (w_{13} = 0.044), percentage of terror (w_{14} = 0.04), percentage of thriller (w_{14} = 0.038), percentage of science fiction (w_{15} = 0.046), and synopsis (w_{16} = 0.08) [16].

Note that the weights shown in brackets for each category have been experimentally obtained using our recommendation algorithm combined with a genetic algorithm (GA). We have performed and experiment where 6 people answered the 70 movies of the preferences test[17]. After extracting the results of the test, we used the data to run an experiment using 60% of the data to train the GA and the 40% left to test the results. Our GA manages a population of vectors of weights (w_k). These vectors are combined and mutated in order to maximize the fitness function. Our fitness function is the Mean Average Error (MAE) where we compare the real rating given by our users to the prediction that the recommender system has given.

To compute the percentage that each movie has of each genre we apply Information Extraction (IE) techniques [63] to the textual synopsis. The IE algorithm searches for key terms that are associated to each genre (after text normalization). Finally the textual synopsis is compared by a cosine distance metric.

3.4. *Group Recommendation module*

Suppose there are n users, $U = \{u : 1 \dots n\}$, let $G_a \subseteq U$ be an active group of users, in our case a group who intend going to see a movie together. The goal is to recommend k items from a set $\{i : 1 \dots k\}$ of target T_a items. We do this by computing a predicted rating $\hat{r}_{G_a,i}$ for active group G_a and each target item $i \in T_a$, and then recommending the k items in T_a that have the highest predicted ratings.

To obtain a prediction for the group we aggregate the predicted ratings of the members, $\hat{r}_{u,i}$ for each $u \in G_a$ for the various i in T_a. Possible aggregation functions include *least misery* (where the minimum is taken), *most pleasure* (where the maximum is taken) or *average satisfaction* (where the average of the predicted ratings of each group member is taken). With the data retrieved in our experiments in simulated environments [10, 11] we have performed a conscientious experimentation comparing the recommendation results of the state-of-the-art aggregation functions presented by [22] when applying them to what we will next define as *standard* and *social* recommendations approaches. The results of these experiments are out of the scope of this paper but can be found in [64]. During this experimentation we have found that *average satisfaction* reported better results for small groups (we consider groups of 10 or less as small) than the other studied aggregation functions, and therefore it is the strategy adopted in *HappyMovie* (as for the moment we do not expect to have large groups using the application):

$$\hat{r}_{G_a,i} \triangleq \frac{1}{|G_a|} \sum_{u \in G_a} \hat{r}_{u,i} \tag{6}$$

We will designate this baseline recommender by *Standard Group Recommender* which will be our state-of-the-art recommender to compare with.

Social group recommender

As we have previously explained, our approach provides an improvement in the accuracy of predicted group ratings by taking into account users' personality and the strength of their connections (which we refer to as their trust). The prediction strategy that takes this extra social knowledge into account is called the *delegation-based rating* method (*dbr*). Using the *average satisfaction* principle again, *Social Recommenders* that use the *dbr* method are defined as:

$$\hat{r}_{G_a,i} \triangleq \frac{1}{|G_a|} \sum_{u \in G_a} dbr(\hat{r}_{u,i}, G_a) \tag{7}$$

Here the *average satisfaction* principle is not applied directly to individual predicted ratings, $\hat{r}_{u,i}$. The ratings are modified by the *dbr* function, which takes into account personality and trust values within the group G_a to compute what we call a *delegation-based rating*. The delegation-based method recognizes that a person's opinions may be based in part on the opinions of other group members. Basically, in each user's turn the user's opinion is not taken into account but it is considered in the other (n-1) turns that is when the user influences others. We know that this is not an intuitive idea. Basically, instead of taking users' opinion once into account, the method takes it several times into account, once for each other user in the group. In our previous work [11, 64], when testing our method in simulated environments, we showed that our delegation-based method improves the accuracy of predicted group ratings more than any other

[16] Extracted genres correspond to the genres used by the web the Web Crawler searches (*http://www.guiadelocio.com*).

[17] We are aware of the limitations of this experiment given the low number of respondents and intend to make further analysis now that we have gathered more information as a result of the experiment carried out in this paper.

standard or *social* approach that we have studied. We here present a refined version of the original formula presented in [11, 64]. The formula, which we explain below, is as follows:

$$dbr(\hat{r}_{u,i}, G_a) = \frac{1}{T} \sum_{v \neq u \in G_a} t_{u,v} [\, \hat{r}_{v,i} + \theta_{r_{v,i}} \cdot \Delta p_{u,v} \,] + m_v \quad (8)$$

where

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

$$\Delta p_{u,v} = p_v - p_u$$

$$m_v = \alpha(1 - s_v) p_v;$$

In Equation 8, $t_{u,v}$ denotes the trust between u and v, which is a real number between 0.0 (no connection) and 1.0 (strong connection). For a given user u in group G_a, we take into account the predicted ratings, $\hat{r}_{v,i}$, for the rest of the group members, $v \in G_a, v \neq u$, weighted by the trust between the two users, $t_{u,v}$. This follows [21], where a method for group recommendations using trust is proposed.

Variable p_u denotes user u's personality, also a real number between 0.0 (very cooperative) and 1.0 (very selfish). The predicted rating of the other group members $\hat{r}_{v,i}$ is increased or decreased depending on the difference in personalities, $\Delta p_{u,v}$. This way, users with stronger personalities will contribute more to the final score.

In this paper (differently to [11, 64]) we have included a $\theta_{r_{v,i}}$ factor. We believe that when modifying a user's predicted preference, $\hat{r}_{u,i}$, for an item i according to trusted friends' preferences, $\hat{r}_{v,i}$, (this is *dbr*'s goal) it is necessary to acknowledge whether the preference of the trusted friend, $\hat{r}_{v,i}$, is positive or negative with respect to the questioned item i. Meaning that a user v with a positive opinion of i, i.e. where $\hat{r}_{v,i}$ is greater than the mid-point of the ratings scale, will want to increase u's opinion of i; but if v has a negative opinion, i.e. where $\hat{r}_{v,i}$ is less than the mid-point of the scale, then v will want to decrease u's opinion. We now model this through a function θ:

$$\theta_{r_{v,i}} \hat{=} \begin{cases} 5 & \text{if } \hat{r}_{v,i} \geq mid \\ -5 & \text{otherwise} \end{cases} \quad (9)$$

where *mid* is the mid-point of the ratings scale, e.g. 3 on a five-point Likert scale. We have chosen constants 5 and -5 because after several studies in group personality composition [13, 11] we have observed that the mean difference in group personality composition is 0.2 and therefore the impact of $\theta_{r_{v,i}} \cdot \Delta p_{u,v}$ in Equation 8 will typically be 1 or -1, which in comparison with other tested ranges has proven to be the most adequate.

Finally, we now include m_v, that represents the memory of past recommendations. The satisfaction value s_v is the level of satisfaction of user v, as explained in Section 3.2. Note that initially all users are assigned a $s_v = 1$. Therefore, the first time that a group receives a recommendation the memory factor is nullified in the formula as it is not necessary because there are not previous recommendations. Parameter α is used to modify the impact of memory in *dbr*. It has a positive or negative value according to $\hat{r}_{v,i}$ in the same way that $\theta_{r_{v,i}}$ has. In the experiments carried out in Section 4.4 we have considered $\alpha = 1$ to get a clearer picture of the impact of the satisfaction value. It is important to note that this satisfaction value is also weighted depending on user v's personality to reflect the importance of satisfying that concrete user. Once the recommendation process has finished the s_v value is updated for every user. Note that here we have intentionally omitted the time-stamp (t) for the sake of readability.

The recommender recommends the k items i from T_a for which $\hat{r}_{G_a,i}$ is highest. We will designate this recommender by *Social Group Recommender*.

4. Experimental Evaluation

In order to verify our *Social Group Recommender* method and *HappyMovie*'s usability we have run an experiment with real users testing our application. Concretely, we have performed four different evaluations: E_1) a functional evaluation of the application to validate its performance from the users' point of view, E_2) a conceptual evaluation to validate the improvement of using *social recommenders* versus *standard recommenders*, E_3) a conceptual evaluation to validate the necessity of using recommendations that take into account users' satisfaction with past recommendations and E_4) a descriptive analysis of the social factors that enhance the recommendations.

We managed to gather 60 users (25 females and 35 males) that completed the whole experiment. Users are students in their twenties from an AI course. All participants used Facebook regularly and had been members for at least one year.

In order to make a further analysis of the results we have considered three different stratified analysis according to personality (strong or mild), genre (male or female) and trust (high or low). However, this last analysis (trust) cannot be performed because it is not an individual feature such as personality and it changes in

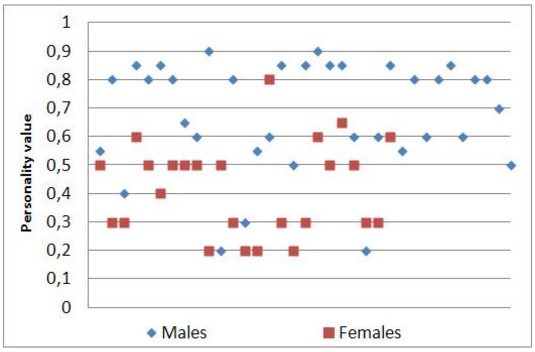

Figure 7: Distribution of users's personality according to their genre

relation to each pair of group members[18]. Therefore we set-up the comparison of the following subpopulations: users whose personality score (p_u) is higher than 0.6 and are therefore considered as "Strong Personality" users (32 users fell into this fold) compared to users whose personality score (p_u) is lower than 0.6 and are therefore considered as "Mild Personality" users (28 users fell into this fold) and males compared to females.

Doing this stratified analysis we have found a correlation between genre and personality. We have observed (as shown in Figure 7) that male students tend to have higher personality values (average 0.677), implying more selfish personalities (74,28% of the males fell into the strong personality fold), whereas female students have a lower average personality value (0.422), implying less selfish personalities (80% of the females fell into the mild personality fold). We have performed the *ANOVA* test [65] to study the effect that genre has on the personality value and found by refuting the null hypothesis that there is indeed a relation between these two variables[19]. However, as we are not psychological experts we will not draw firm conclusions on this matter, as there might be something in the personality test that we are unaware of, maybe in the way that questions are formulated that leads to these gender differences. Therefore we do not want to extrapolate it to a general population fact and remark it as an intrinsic aspect of this sample.

4.1. E_1) Functional Evaluation

Users were asked to test the functionality offered by the application and answer a questionnaire. More precisely they were asked to perform the following steps:

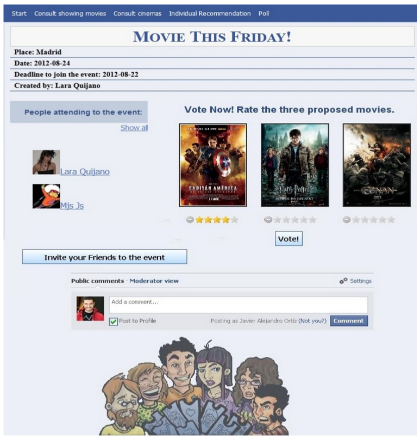

Figure 8: *HappyMovie*'s event page

Step 1. Answer the personality test through the movie metaphor (Figure 3). **Step 2.** Answer the preferences test (Figure 6). Users were asked to rate as many movies as possible, if they had not watched the movie they could skip it. In the end we have gathered 3863 ratings, with an average of 64 ratings per user. Our preferences test allows to distinguish if users don't know a movie, or if they know the movie but have not watched it on purpose because they totally despise it. For example, a user might have seen some *Almodovar*'s movies, and do not intend to see any more because s/he does not like them, giving a score of 0 to the new *Almodovar*'s movie. **Step 3.** Check the accuracy of the recommended movies presented by the "Individual Recommendation" tab. **Step 4.** Meet together grouping themselves in groups of 3 people and create an event to go to the movies together. **Step 5.** Look at the complete current movie listing at local cinemas and debate until there is an agreement which 3 movies they would like to see in a movie outing (users were also asked to individually think which 3 movies they would like to see before performing step 6). **Step 6.** Check the 3 best movies that the application has found for the group[20]. Internally debate whether they would follow or not the recommendation and how satisfied they are with it. Individually rate the presented movies through the 5-star system presented in the event's page (see Figure 8).

Next they answered individually the following questions, with a five star Likert scale[21]:

[18]We could have used the average trust, however, we did not consider that it would be a representative variable as it always tends to similar values (around 0.4) as explained in Section 4.4.

[19]The obtained F-value and p-value are 28.6828 and <0.0001 respectively.

[20]These 3 movies are presented in *HappyMovie*'s event page (see Figure 8), where apart from event related information (guests, celebration place, date and time, etc) the application displays the best 3 retrieved movies for the current group along with a 5-star voting system that allows *HappyMovie* to update users' satisfaction value (s_u).

[21]We ran the experiment with students whose first language was

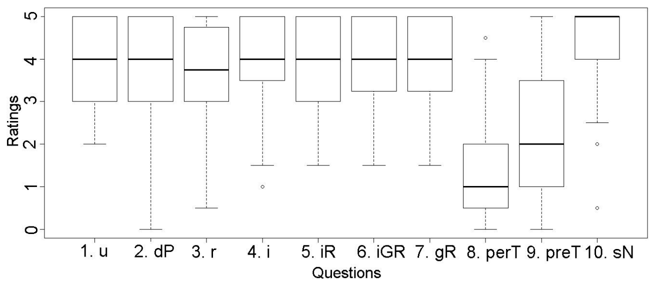

Figure 9: Users answers to *HappyMovie*'s questionnaire

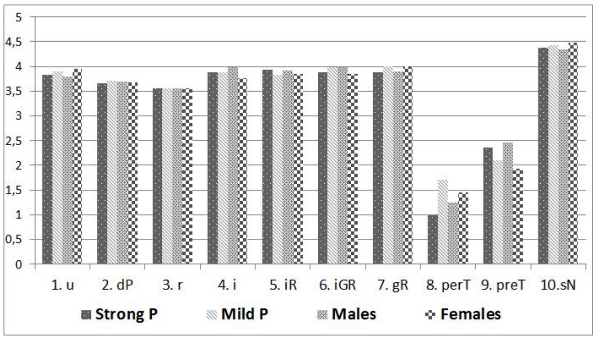

Figure 10: Average of user's answers to *HappyMovie*'s questionnaire. Data analysis comparison.

Q1. Usefulness (u): "*I find the application useful (being 0 not useful at all and 5 very useful)*".

Q2. Decision process (dP): "*It is useful because it speeds up the group decision process (being 0 very little and 5 a lot)*".

Q3. Reusability (r): "*I will use the application to go to the movies with my friends (being 0 very little and 5 a lot)*".

Q4. Usability (i): "*The application is intuitive and easy to use (being 0 not at all intuitive and 5 very intuitive)*".

Q5. Individual Recommendation (iR): "*I like the individual recommendation of the system (being 0 barely and 5 a lot)*".

Q6. Individual Group Recommendation (iGR): "*I individually like the group recommendation of the system (being 0 barely and 5 a lot)*".

Q7. Group Recommendation (gR): "*As a group we like the group recommendation of the system (being 0 barely and 5 a lot)*".

Q8. Personality Test (perT): "*Was it easy to answer to the personality test? (being 0 very easy and 5 not easy at all)*".

Q9. Preferences Test (preT): "*Was it easy to answer to the preferences test? (being 0 very easy and 5 not easy at all)*".

Q10. Social network (sN): "*Do you like having the application in a social network? (being 0 not at all positive and 5 very positive)*".

Figure 9 shows the test's general results and Figure 10 shows the average of the results when analyzing the stratified data. Note that results for users with more selfish personality values are very similar to results for male users; and results for users with less selfish personality values are very similar to results for female users. This follows from the background observation we made, that male students had on average more selfish personalities than female students. When comparing "Strong Personality" users' answers with the "Mild Personality" ones we do not find significant differences. Both subpopulations seem equally (dis)satisfied. We now evaluate users' answers to each question:

Q1. Usefulness (u) [$\overline{u} = 3.86$, $s_u = 0.911$]: Reflects that users' opinion about the application usefulness is very high (being 5 the top value).

Q2. Decision process (dP) Average score [$\overline{dP} = 3.69$, $s_{dP} = 1.20$]: Supports a positive opinion about the speed up of the decision process given by the application.

Q3. Reusability (r) Average score [$\overline{r} = 3.56$, $s_r = 1.23$]: Reflects users' good predisposition to use frequently the application. This inclination is probably motivated by the two previous answers where users expressed that they consider the application useful and that it speeds up their group decision, therefore they intend to return and use it again.

Q4. Usability (i) Average score [$\overline{i} = 3.89$, $s_i = 1.08$]: This high response towards the usability question reflects that users think that the application is intuitive and easy to use without further instructions.

Q5. Individual Recommendation (iR) Average score [$\overline{iR} = 3.89$, $s_{iR} = 1.01$]: Reflects users' happiness with their individual recommendations.

Q6. Individual Group Recommendation (iGR) Average score [$\overline{iGR} = 3.93$, $s_{iGR} = 1.06$]: Reflects that users are individually satisfied with the group recommendations.

Q7. Group Recommendation (gR) Average score [$\overline{gR} = 3.94$, $s_{gR} = 0.921$]: Shows that users think that the application made a good group recommendation. We consider that this is a good result towards our *Social Group Recommendation Method* as this is one of the questions with the highest scoring.

Spanish. The questions that we show here are paraphrases into English of the Spanish questionnaire.

Q8. Personality Test (perT) Average score [$\overline{perT}$ = 1.33, s_{perT} = 1.07]: This value is on average not very high which is good, because it means that users do not resent doing the personality test. With the interactive metaphor we have managed to make the application more usable and entertaining.

Q9. Preferences Test (preT) Average score [$\overline{preT}$ = 2.24, s_{preT} = 1.33]: This value is by far the worst result in the questionnaire. We consider that although it is not high enough (more than 3) to represent that users resent answering the preferences test it is quite a high value. However, we decided that it was worth sacrificing this question's results, asking users to rate a lot of movies in the preferences test (70 movies), in order to offer better recommendations (as we have explained in Section 3.3 the more precise the individual profile is the better the recommenders perform).

Q10. Social network (sN) Average score [$\overline{sN}$ = 4.41, s_{sN} = 0.9]: From this answer we can conclude that users totally approve having the application in a social network.

4.2. *E_2) Social recommender vs. Standard recommender*

We have also tested whether social factors improve the performance of group recommendations. This premise, that was proven to be true in our experiments with simulated environments [10, 11, 64], has been now also confirmed when using our *Social Group Recommendation Method* through *HappyMovie*[22].

Once every group had answered the test, we replaced the *Social Group Recommender* (Equation 7) by the *Standard Group Recommender* (Equation 6). Users were asked to repeat steps 5 and 6 of the experiment and answer again questions *Q6. Individual Group Recommendation* and *Q7. Group Recommendation*[23]. Figure 11 shows the comparison between answers to both questions with and without the inclusion of social factors in the group recommendation. As we can see users' opinion about the group recommendation, individually and as a group, is far higher when *HappyMovie* has a *Social Group Recommender* configuration than when it has a *Standard Group Recommender* configuration. Statistical significance tests have been carried out to prove that these differences are significant. We have used Wilcoxon test [66] as our sample is not normally distributed. Firstly we have proven that there are no dependencies between both answers and that they do not follow a random pattern by choosing as null hypothesis that "the ordinal relationships among the measures are equal to". We have obtained p-values < 0.05 (6.886e-06 for Social vs Standard comparing question's *Q6* results and 6.251e-06 for Social vs Standard comparing question's *Q7* results). This proves (as Wilcoxon test assures) that when we compare two rated samples and the p-value is lower than 0.05 we can assure that the two series are different, are not a result of chance and are said to be statistically significant. We have performed a variation of Wilcoxon test, this time, proving by choosing the null hypothesis as "less than" that *Standard Group Recommendation* results are smaller than the *Social Group Recommendation* ones. With this test we have also obtained p-values < 0.05 (3.443e-06 for Social vs Standard comparing question's *Q6* results and 3.125e-06 for Social vs Standard comparing question's *Q7* results). Therefore we can conclude that indeed our *Social* method improves the performance of other group recommendations that do not use social factors.

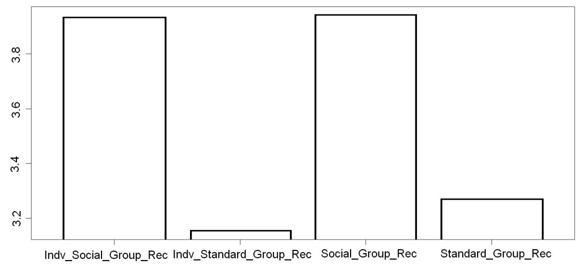

Figure 11: Users reaction to *Standard* and *Social Group Recommenders*

4.3. *E_3) Conceptual evaluation of the long term recommendation module*

Next, we have tested the impact of users' opinion through time. This was motivated by the idea that, as reflected in answer *Q3. Reusability* from our questionnaire, users would like to use regularly our application when going to the movies. To do so, we asked users to return one month later and use the application again with the same group configuration. Consequently, cinemas' movie listings had changed. As we explained in

[22] Note that in our previous experiments we performed severe testing comparing our *Social Group Recommendation Method* with several state-of-the-art group recommenders that did not use social factors. In this paper we will just limit to confirm our previous conclusions by comparing our results with a state-of-the-art recommender that uses an average satisfaction approach as explained in Equation 6.

[23] Note that users were not explained at any moment the concepts of social or standard recommenders, the purpose of our experiment or that we had changed the recommendation methods. Users were just asked to check again for the movies that the recommender proposed and answer whether they liked this proposal better or worse than the previous one by answering again to questions Q6 and Q7.

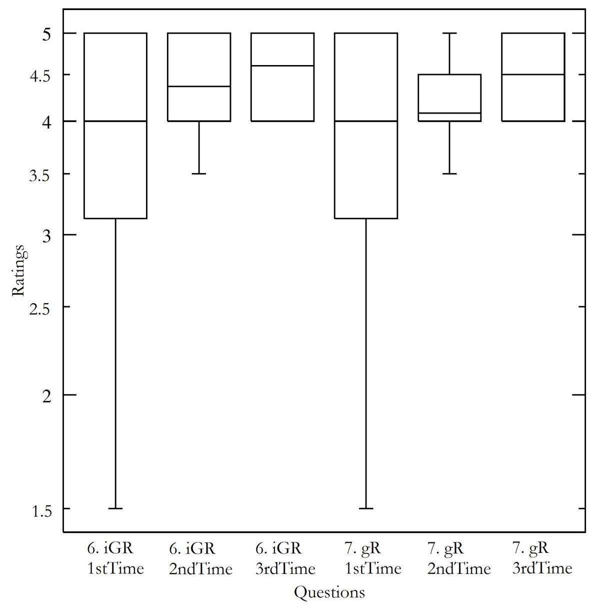

Figure 12: Users reaction to our group recommendations over time

Results	p-value "equal to" n.h	p-value "less than" n.h
1stTime vs. 2ndTime Indv	0.04654	0.02327
1stTime vs. 2ndTime Group	0.00135	0.0006751
2ndTime vs. 3rdTime Indv	0.001715	0.0008573
2ndTime vs. 3rdTime Group	0.001715	0.0008573

Table 1: p-value results for Wilcoxon test

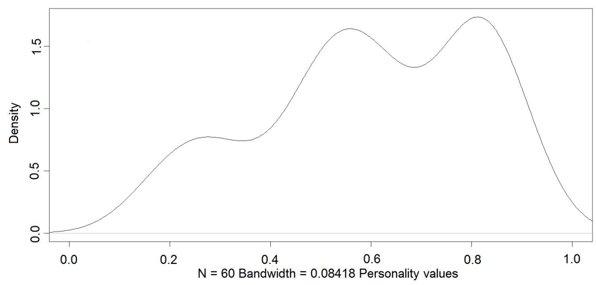

Figure 13: Density distribution of the personality factor

Sections 3.2 and 3.4 our group recommendation method keeps a memory of past recommendations and tries to ensure a balanced user satisfaction. We asked users to repeat again steps 5 and 6 of the experiment and answer to questions *Q6. Individual Group Recommendation* and *Q7. Group Recommendation*, that were now slightly modified for its better adjustment to this part of the experiment as follows:

Q6. Individual Group Recommendation (iGR): "*I am individually satisfied with the group recommendation of the system (being 0 barely and 5 a lot)*".

Q7. Group Recommendation (gR): "*We are satisfied as a group with the group recommendation of the system (being 0 barely and 5 a lot)*".

One month later users were asked to return again (so that cinemas' movie listings had changed again), repeat the 5th and 6th step of the experiment and answer for the third time questions *Q6* and *Q7* from our questionnaire. Figure 12 shows users' satisfaction with our group recommendation individually and as a group. Looking at users' answers, both individually and as a group, users' satisfaction with the recommendations keeps improving reaching out in the third time values higher than 4.5. In the 2nd and 3rd time users' average individual satisfaction is higher than group satisfaction. This can be due to a feeling that the recommender has favoured them, or that they have "won" in the decision about which movie the group will watch. This is not a surprising result as it was our goal (as explained in Section 3.2) to favour those users less satisfied with the previous recommendation through the satisfaction value (s_u). Regarding this fact, the most important conclusion is that we can appreciate that there are no dissatisfied users the 2nd time and that in the 3rd time not only there are no dissatisfied users but also only highly satisfied ones. Hence, we can conclude that it is worth having a system that contemplates users' satisfaction over time and tries to ensure a general global satisfaction as results show users' increasing well-fare. These results have been statistically confirmed through Wilcoxon test [66] obtaining p-value results less than 0.05, as shown in Table 1. As it can be seen, we have tested the two different null hypothesis (n.h) "equal to" (that proves that results are different and are not a result of chance and therefore statistically significant) and "less than" (that proves that the differences in the results are statistically significant).

Moreover we have performed numerical analysis of the personality and trust factors to find correlations and study their impact on the recommendation method.

4.4. *E_4) Descriptive analysis of the social factors*

Analysis of the personality factor: We have analyzed users' answers to the given personality test. The average p_u is 0.57 and the p_u values distribution is reflected in Figure 13 where the standard deviation is 0.221. We can conclude from this analysis than on average users have high personality values, as the majority of them are comprehended in the [0.5,0.8] range. However the variance in users' personality is high, we have users with very strong personalities (0.9) and very mild personalities (0.2).

Analysis of the trust factor: Once our experiment was over, we analyzed the trust between each group member and found a lot of diversity in the data. The average trust is 0.41 and the standard deviation is 0.259. Different users had completely opposite levels of trust with values of 0.86 or 0.10. This means that our sample has a varied representation of relationships, some of them are just classmates and others are close friends[24]. On average trust is not very high, we consider this aspect to be predictable due to the sample of people taken. For example, when testing the obtained improvement in the recommendation accuracy when using our *Social Group Recommender* method in simulated environments [10, 11] our data was formed by our friends in two different Social Networks thus the average trust was higher (0.597). We have also studied if trust was related to personality, and for example groups with strong or mild personality on average had strong or weak trust between them. However, we have not found any evidence of this. From this study we can conclude that trust between users is related to each person individually and has nothing to do with the personality of each individual. We have found very "trusting" users with strong and mild personalities and the other way round.

5. Conclusions and Future Work

This paper extends our previous work regarding *Social Group Recommenders* [10, 11, 64] and presents a comprehensive description of *HappyMovie*, our social recommender application. We provide a complete description of the system that may serve to reproduce the proposed techniques in other recommender systems. To illustrate the advantages of the approach, a functional description and evaluation is presented.

Through the inclusion of social factors –namely personality and trust– the *HappyMovie* system can ease the real decision making process performed by groups of people when choosing a movie to watch together. The simulation of this process is implemented through different modules that obtain and provide social knowledge, estimate the individual and group preferences, and include a long term knowledge management regarding users satisfaction with previous recommendations. Throughout the different experiments presented in this paper we have proven users' acceptance towards the system and tested the higher acceptance of our *Social Group Recommendation Method* proposals compared to the ones provided by the *Standard Group Recommender*. We have also justified the need for considering a system that takes into account previous group recommendations events by evaluating the global welfare and satisfaction of users through time.

Users' answers to our different questionnaires have reflected that they are willing to use *HappyMovie*. Some of the reasons for this positive response are that users believe that *HappyMovie* easies group decisions and that it is easy to use. But the most interesting and important feedback that users have given us is that they like the individual and group recommendations that *HappyMovie* offers and that this positive opinion increases the more they use the application. Therefore we believe that it is worth making the application public.

One extra advantage of building *HappyMovie* has been obtaining data, such as ratings, personality values, etc. This was a very difficult and costly matter when we had our recommendation method embedded in a standalone system [10, 11]. With *HappyMovie* we have been able to extract automatically most of the data required by our system (for example the trust factor) and also to obtain a bigger sample of data (we now count with a database of 3863 ratings). We will now be able to conduct further experiments using the data obtained and expect to obtain more if users start using *HappyMovie* every time they need a movie recommendation.

Although our *social recommendation* approach has been applied to the movies domain, it can be reproduced in other domains as *HappyMovie* follows a generic architecture called ARISE and a development process based on software templates has also been provided [12]. As future work we would like to obtain data enough to carry Big Data experiments, provide a public group recommender dataset, and also perform further group analysis, like measuring how dissimilar or similar are preferences within a group, diversity, serendipity, etc. Another interesting on-going work is the inclusion of explanations to users, through them the system will try to justify the proposed items and increase users' level of acceptance by displaying others' needs.

[24] This kind of limitation when having to find a sample of objective users is a problem that most researchers find when testing group recommenders [6].

References

[1] G. Adomavicius, A. Tuzhilin, Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749.

[2] A. Jameson, B. Smyth, Recommendation to Groups, in: The Adaptive Web, Methods and Strategies of Web Personalization, Lecture Notes in Computer Science, Springer, ISBN 978-3-540-72078-2, 596–627, 2007.

[3] L. Baltrunas, T. Makcinskas, F. Ricci, Group recommendations with rank aggregation and collaborative filtering, in: RecSys '10, ACM, New York, NY, USA, 119–126, 2010.

[4] S. Berkovsky, J. Freyne, Group-based recipe recommendations: analysis of data aggregation strategies, in: RecSys, 111–118, 2010.

[5] T. D. Pessemier, S. Dooms, L. Martens, Comparison of group recommendation algorithms, in: Multimedia Tools an Applications, 2013.

[6] J. Masthoff, A. Gatt, In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems, User Modeling and User-Adapted Interaction (2006) 281–319.

[7] C. A. Erdley, M. Rivera, E. Shepherd, L. J. Holleb, Social-cognitive models and skills, in: Practitioners guide to empirically based measures of social skills, Springer, 21–35, 2010.

[8] N. A. Christakis, J. H. Fowler, Social Contagion Theory: Examining Dynamic Social Networks and Human Behavior, CoRR abs/1109.5235.

[9] M. Bazire, P. Brézillon, Understanding Context Before Using It, in: Context, 29–40, 2005.

[10] L. Quijano-Sánchez, J. A. Recio-García, B. Díaz-Agudo, Personality and Social Trust in Group Recommendations, in: ICTAI'10., IEEE Computing Society, 121–126, 2010.

[11] L. Quijano-Sánchez, J. A. Recio-García, B. Díaz-Agudo, G. Jiménez-Díaz, Social factors in group recommender systems, in: ACM TIST, vol. 4, 8, 2013.

[12] L. Quijano-Sánchez, J. A. Recio-García, B. Díaz-Agudo, A Reusable Methodology for the Instantiation of Social Recommender Systems, in: ICTAI'13, in press, 2013.

[13] L. Quijano-Sánchez, J. A. Recio-García, B. Díaz-Agudo, HappyMovie: A Facebook Application for Recommending Movies to Groups, in: ICTAI'11, 239–244, 2011.

[14] J. McCarthy, T. Anagnost, MusicFX: An Arbiter of Group Preferences for Computer Aupported Collaborative Workouts, in: CSCW '98, ACM, 363–372, 1998.

[15] A. Crossen, J. Budzik, K. Hammond, Flytrap: intelligent group music recommendation, in: IUI '02, ACM, 184–185, 2002.

[16] M. O'Connor, D. Cosley, J. Konstan, J. Riedl, PolyLens: a recommender system for groups of users, in: ECSCW'01, ISBN 0-7923-7162-3, 199–218, 2001.

[17] J. McCarthy, Pocket Restaurant Finder: A situated recommender systems for groups, in: Procs. in Workshop on Mobile Ad-Hoc Communication at CHI'02, ACM, 2002.

[18] A. Jameson, More than the sum of its members: challenges for group recommender systems, in: AVI '04, ACM, 48–54, 2004.

[19] J. Golbeck, Combining Provenance with Trust in Social Networks for Semantic Web Content Filtering, in: IPAW'06, Lecture Notes in Computer Science, Springer, ISBN 3-540-46302-X, 101–108, 2006.

[20] R. Sinha, K. Swearingen, Comparing Recommendations Made by Online Systems and Friends, in: DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries, 2001.

[21] J. Golbeck, Generating Predictive Movie Recommendations from Trust in Social Networks, in: iTrust: 4th International Conference on Trust Management, 93–104, 2006.

[22] J. Masthoff, Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers, User Model. User-Adapt. Interact. 14 (1) (2004) 37–85.

[23] C. Baccigalupo, E. Plaza, A Case-Based Song Scheduler for Group Customised Radio, in: ICCBR, Belfast, 2007, Proceedings, vol. 4626 of *Lecture Notes in Computer Science*, Springer, ISBN 978-3-540-74138-1, 433–448, 2007.

[24] I. Konstas, V. Stathopoulos, J. M. Jose, On social networks and collaborative recommendation, in: SIGIR, 195–202, 2009.

[25] W.-P. Lee, C. Kaoli, J.-Y. Huang, A smart TV system with body-gesture control, tag-based rating and context-aware recommendation, Knowledge-Based Systems 56 (2014) 167–178.

[26] H. Lieberman, N. W. V. Dyke, A. S. Vivacqua, Let's Browse: A Collaborative Web Browsing Agent, in: IUI, 65–68, 1999.

[27] X. H. Jiliang Tang, H. Liu, Social recommendation: a review, in: Social Network Analysis and Mining, Springer, 2013.

[28] M. Gartrell, X. Xing, Q. Lv, A. Beach, R. Han, S. Mishra, K. Seada, Enhancing group recommendation by incorporating social relationship interactions, in: GROUP, 97–106, 2010.

[29] S. Huanga, X. Sub, Y. Huc, S. Mahadevand, Y. Deng, A new decision-making method by incomplete preferences based on evidence distance, Knowledge-Based Systems 56 (2014) 264–272.

[30] A. Mislove, B. Viswanath, P. K. Gummadi, P. Druschel, You are who you know: inferring user profiles in online social networks, in: WSDM, 251–260, 2010.

[31] N. Ma, E.-P. Lim, V.-A. Nguyen, A. Sun, H. Liu, Trust relationship prediction using online product review data, in: CIKM-CNIKM, 47–54, 2009.

[32] M. Jamali, M. Ester, *TrustWalker*: a random walk model for combining trust-based and item-based recommendation, in: KDD, 397–406, 2009.

[33] F. Zhang, L. Bai, F. Gao, A User Trust-Based Collaborative Filtering Recommendation Algorithm, in: ICICS, 411–424, 2009.

[34] S. Nepal, C. Paris, S. K. Bista, SRec: a social behaviour based recommender for online communities, in: UMAP Workshops, 2012.

[35] M. S. Pera, N. Condie, Y.-K. Ng, Personalized Book Recommendations Created by Using Social Media Data, in: WISE Workshops, 390–403, 2010.

[36] X. Yang, H. Steck, Y. Guo, Y. Liu, On top-k recommendation using social networks, in: RecSys, 67–74, 2012.

[37] J. Bao, Y. Zheng, M. F. Mokbel, Location-based and preference-aware recommendation using sparse geo-social networking data, in: SIGSPATIAL/GIS, 199–208, 2012.

[38] X. Hu, C. Chen, X. Chen, Z.-K. Zhang, Social Recommender Systems Based on Coupling Network Structure Analysis, CoRR abs/1204.1949.

[39] U. Shardanand, P. Maes, Social Information Filtering: Algorithms for Automating "Word of Mouth", in: CHI, 210–217, 1995.

[40] IBM, Ibm's black friday report says twitter delivered 0 percent of referral traffic and facebook sent just 0.68 percent, in: https://strme.wordpress.com/2012/11/27/ibmsblack-friday-report-says-twitter-delivered-0-percent-of-referral-traffic-and-facebook-sentjust-0-68-percent/, 2012.

[41] Quora, Why does the startup idea of social recommendations consistently fail?, in: http://www.quora.com/Why-does-the-startup-idea-of-social-recommendationsconsistently-fail, 2012.

[42] Y. Chen, L. Cheng, C. Chuang, A group recommendation system with consideration of interactions among group members, Expert Systems Applications (2008) 2082–2090ISSN 0957-4174, doi:http://dx.doi.org/10.1016/j.eswa.2007.02.008.

[43] K. Bischoff, Exploiting Social Ties for Search and Recommendation in Online Social Networks - Challenges and Chances, in: Grundlagen von Datenbanken, 2010.

[44] Merriam-Webster's Collegiate Dictionary 10th Ed, 2002.

[45] H. Fujita, J. Hakura, M. Kurematsu, Intelligent human interface based on mental cloning-based software, Knowl.-Based Syst. 22 (3) (2009) 216–234.

[46] A. Minamikawa, H. Fujita, J. Hakura, M. Kurematsu, Personality Estimation Application for Social Media, in: SoMeT, 327–335, 2012.

[47] K. Thomas, R. Kilmann, Thomas-Kilmann Conflict Mode In-

strument, Tuxedo, N.Y., 1974.
[48] E. Berne, Games People Play The Basic Hand Book of Transactional Analysis, New York: Ballantine Books, 1964.
[49] P. T. Barrett, K. Petrides, S. B. Eysenck, H. J. Eysenck, The Eysenck Personality Questionnaire: An examination of the factorial similarity of P, E, N, and L across 34 countries, Personality and Individual Differences 25 (5) (1998) 805–819.
[50] J. Paul T. Costa, R. R. McCrae, Domains and Facets: Hierarchical Personality Assessment Using the Revised NEO Personality Inventory, Journal of Personality Assesment 64(1) (1995) 21–50.
[51] R. R. McCrae, 0liver P. John, An introduction to the five-factor model and its applications, Journal of Personality 60 (1992) 175–215.
[52] J. L. Herlocker, J. A. Konstan, L. G. Terveen, J. T. Riedl, Evaluating collaborative filtering recommender systems, ACM Trans. Inf. Syst. 22 (1) (2004) 5–53, ISSN 1046-8188, doi: http://doi.acm.org/10.1145/963770.963772.
[53] N. A. Schaubhut, Technical Brief for the Thomas-Kilmann conflict mode instrument, CPP Research Department, 2007.
[54] J. A. Recio-García, G. Jimenez-Diaz, A. A. Sánchez-Ruiz, B. Díaz-Agudo, Personality aware recommendations to groups, in: Procs. of the 2009 ACM Conference on Recommender Systems, ACM, 325–328, 2009.
[55] P. Avesani, P. Massa, R. Tiella, A trust-enhanced recommender system application: Moleskiing, in: SAC '05, ACM, NY, USA, 1589–1593, 2005.
[56] E. Gilbert, K. Karahalios, Predicting tie strength with social media, in: CHI '09, ACM, 211–220, 2009.
[57] D. Z. Levin, R. Cross, L. C. Abrams, The strength of weak ties you can trust: the mediating role of trust in effective knowledge transfer, Management Science 50 (2004) 1477–1490.
[58] M. Granovetter, The Strength of Weak Ties, The American Journal of Sociology 78 (6) (1973) 1360–1380.
[59] Y.-G. Cho, K.-T. Cho, A loss function approach to group preference aggregation in the AHP, Computers & OR 35 (3) (2008) 884–892.
[60] M. Volkovs, R. S. Zemel, A flexible generative model for preference aggregation, in: WWW, 479–488, 2012.
[61] P. Lops, M. de Gemmis, G. Semeraro, Content-based Recommender Systems: State of the Art and Trends, in: Recommender Systems Handbook, 73–105, 2011.
[62] M. Ekstrand, J. Riedl, J. Konstan, Collaborative Filtering Recommender Systems, Foundations and Trends in Human-Computer Interaction 4 (2) (2011) 175–243.
[63] S. Sarawagi, Information Extraction, Foundations and Trends in Databases 1 (3) (2008) 261–377.
[64] L. Quijano-Sánchez, J. A. Recio-García, B. Díaz-Agudo, Group recommendation methods for social network environments, in: 3rd Workshop on Recommender Systems and the Social Web, RecSys, 383–384, 2011.
[65] D. C. Howell, Statistical methods for psychology (5th ed.), Pacific Grove, CA: Duxbury/Thomson Learning, ISBN 0-534-37770-X, 2002.
[66] F. Wilcoxon, Individual Comparisons by Ranking Methods, in: Biometrics Bulletin, 1, 80–83, 1945.