A group recommender system for tourist activities

Inma Garcia, Laura Sebastia, Eva Onaindia, and Cesar Guzman

Dpt. Computer Science, Technical Univ. of Valencia
Camino de Vera s/n, 46022, Valencia, Spain
{ingarcia,lstarin,onaindia,cguzman}@dsic.upv.es

Abstract. This paper introduces a method for giving recommendations of tourist activities to a group of users. This method makes recommendations based on the group tastes, their demographic classification and the places visited by the users in former trips. The group recommendation is computed from individual personal recommendations through the use of techniques such as aggregation, intersection or incremental intersection. This method is implemented as an extension of the *e-Tourism* tool, which is a user-adapted tourism and leisure application, whose main component is the *Generalist Recommender System Kernel (GRSK)*, a domain-independent taxonomy-driven search engine that manages the group recommendation.

Key words: Recommender Systems, Group Recommenders, Tourism

1 Introduction

Recommender systems (RS) are widely used in the internet for suggesting products, activities, etc. These systems usually give a recommendation for a single user considering his/her interests and tastes. However, many activities such as watching a movie or going to a restaurant, involve a group of users. In such a case, RS should take into account the likes of all the group users by combining the tastes and preferences of all the members into a single set of preferences and obtain a recommendation as if they were a single user. This tedious and complicated task requires the group members previously agree on the way their particular preferences will be gathered together into a single group profile. In order to overcome this shortcoming, some RS take into account the interests and tastes of the group as a whole. The first task of this type of systems is to identify the individual preferences and then find a compromise that is accepted by all the group members. This is the crucial point in a group RS because how individual preferences are managed to come up with group preferences will determine the success of the recommendation.

This paper is focused on a RS for tourism. *e-Tourism* [8] is a web-based recommender system that computes a user-adapted leisure and tourist plan for both a single user and a group. The system does not solve the problem of travelling to an specific place but it recommends a list of activities that a single tourist

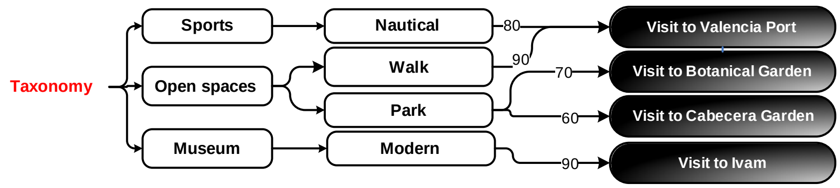

Fig. 1. GRSK Taxonomy.

or group of tourists can perform in a city, particularly, in the city of Valencia (Spain). It also computes a time schedule for the list of recommended activities taking into account the distances between places, the opening hours, etc. - that is, an agenda of activities.

The component of *e-Tourism* in charge of generating the list of activities that are likely of interest to the single user or group of users is the *Generalist Recommender System Kernel (GRSK)*, whose main aspects are explained in Section 2. Section 3 details the basic recommendation techniques used to model the individual user preferences, and section 4 introduces the techniques to compute the final group recommendations. Section 5 presents the experimental setup to evaluate our approach. Section 6 summarizes similar state-of-the-art RS and we finish with some conclusions and future work.

2 The Generalist RS Kernel (GRSK)

The task of the *Generalist Recommender System Kernel (GRSK)* is to generate the list of activities to recommend to a single user or to a group of users. This section describes the main aspects of the GRSK when working as a group RS.

2.1 GRSK Taxonomy

The GRSK behaviour relies on the use of a taxonomy to represent the user's likes and the items to recommend. It has been designed to be *generalist*, that is independent of the current catalog of items to recommend. Therefore, the GRSK is able to work with any application domain as long as the data of the new domain are defined through a taxonomy representation.

The entities in a **taxonomy** are arranged in a hierarchical structure connected through a *is-a* relationship in which the classification levels become more specific towards the bottom. In the GRSK taxonomy (an example is shown in figure 1), entities represent the **features** (F) that are commonly managed in a tourism domain like 'Open Spaces', 'Museums', 'Nautical Sports', etc. as figure 1 shows. The leaf nodes of the taxonomy represent the **items** to recommend; they are categorized by the lowest-level or most specific feature in the hierarchy. The edges linking an item to a feature are associated a value to indicate the **degree of interest** of the item (activity in the tourism taxonomy) under the feature, i.e. as a member of the category denoted by the feature. An item can

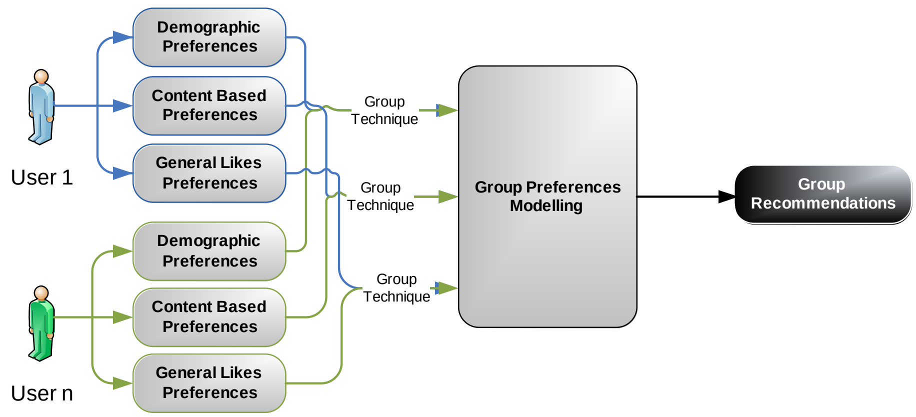

Fig. 2. Group Recommendation Process.

also be categorized by more than one feature in the taxonomy. For instance, in figure 1, the item 'Visit to Valencia Port' is categorized as 80% of interest as 'Nautical Sport' and 90% of interest as a place for going for a 'Walk'.

Items are described by means of a list of tuples which represent all their incoming edges. Each tuple is of the form (i, f, r), where i is the item, $f \in F$ is a feature defined in the taxonomy and r is the degree of interest of the item i under f. Additionally, items are associated a numeric value AC^i (**acceptance counter**) to represent how popular the item i is among users; this value indicates how many times the item i has been accepted when recommended.

2.2 User Information

The GRSK records a **profile** of each individual user u, which contains personal and demographic details like the age, the gender, the family or the country. The profile also keeps information about the **general likes** of the user, denoted by GL^u, which are described by a list of pairs on the form (f, r), where $f \in F$ and $r \in [0, 100]$. A user profile in GRSK also contains information about the historical interaction of the user with the RS, namely the set of items the user has been recommended and his/her degree of satisfaction with the recommendation.

The first step to utilize the system is to register and make up the user profile. Whenever a person asks for a new recommendation, his/her user profile is updated to better capture his/her likes.

A **group of users** G is composed of a set of users already registered in the system. The GRSK takes into account each individual user profile to return the list of recommended items to the group of users.

2.3 Group Recommendation Process

Figure 2 outlines the process for computing the group recommendation. Once the individual preferences of each user are modeled, they are combined to obtain

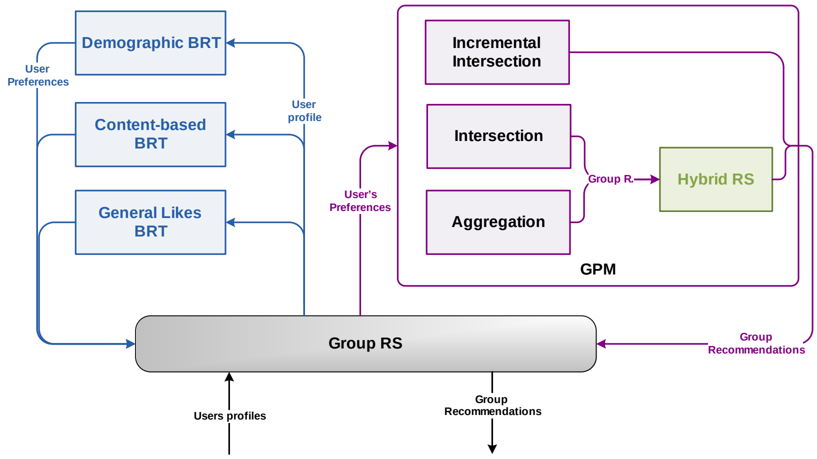

Fig. 3. GRSK Architecture.

the group preferences by means of a group preference modelling. These group preferences are then used to retrieve the list of items to recommend (group recommendations). The individual user preferences as well as the group preferences are described by means of a list of tuples of the form $(u/G, f, r)$, where $f \in F$ and $r \in [0, 100]$.

The main components of the GRSK are shown in figure 3. The Group RS is in charge of controlling the whole recommendation process. First, users profiles are sent by the Group RS to the basic recommendation techniques (BRT) modules that produce a list of individual preferences according to each type of recommendation (demographic RS [3], content-based RS [3] and likes-based filtering [4]). The result of this phase is a set of three lists of individual preferences for each member in the group. These individual preferences are sent by the Group RS to be processed by the Group Preference Manager (GPM), which, through methods like aggregation, intersection or incremental intersection, combines the individual preferences and reports the final group recommendation. The first phase is detailed in the next section whereas the second phase is explained in section 4.

3 Modelling the Individual User Preferences

The first task of the Group RS is to elicit the individual preferences from the users profiles. These preferences are computed by means of three basic recommendation techniques (BRT): demographic and content-based recommendation techniques, and likes-based filtering technique. The preferences returned by each BRT are independent from each other, and they will be later used to recommend the group items.

The **demographic BRT** classifies the user u into a demographic category according to his profile details. For example, a person with children is classified into a different category than a retiree as they will likely have different likes. We will call P_d^u the set of preferences generated by a demographic BRT. We opted for a demographic BRT because it is able to always give a recommendation for the problem of the *new user*. In addition, it can also suggest items other than items previously recommended.

The **content-based BRT** computes a set of preferences by taking into account the items that have been previously rated by the user (historical interaction). We will call P_{cb}^u the set of preferences generated by a content-based BRT. Let f be a feature and I a list of items described by a pair (i, f, r^i) in the taxonomy. Given a user u who has rated a set of items I^u with a value ur^i, a preference (u, f, r^u) is added to P_{cb}^u with:

$$r^u = \frac{\sum\limits_{\forall i \in I \cap I^u} ur^i * r^i}{|I^u|}$$

The value r^u denotes the interest-degree of a user u for the items described under the feature f among the whole set of items rated by u. The use of a content-based BRT will allow us to increase the user satisfaction by recommending items similar to the ones already accepted by the user. For example, if the user likes visiting museums, the system will tend recommending visits to other museums.

The **likes-based filtering module** is an information filtering technique that works with the general user likes GL^u specified by the user in his profile. In this case, the set of preferences P_{lf}^u is simply built as $P_{lf}^u = \{(u, f, r^u) : \forall (f, r) \in GL^u : r^u = r\}$.

4 Generating the Group Recommendations

Once the individual preferences are elicited from the BRT modules, the group RS sends them to the **Group Preference Manager** (GPM) to get the group preferences model (see figure 3). The GPM makes use of three disjunctive methods to construct the group preferences: aggregation, intersection and incremental intersection. These three methods differ in how the lists of individual preferences are combined.

The aggregation mechanism is a common technique that has been used in various group RS (see section 6). This technique gathers the individual preferences of all the group members to make up a single set of preferences. However, aggregating preferences does not necessarily account for the preferences of the group as a whole; the intersection mechanism is thereby introduced as a counterpoint. The intersection technique obtains a set of preferences that are shared by all the participants in the group. The risk of using this mechanism is that we might end up with an empty intersection list if the group is rather heterogeneous. Finally, the incremental intersection mechanism combines the advantages of aggregation and intersection in a single strategy.

The GPM is fed with three lists of individual preferences and builds a list of group preferences calculated with the selected disjunctive method. Individual preferences are denoted by (P_d^u, P_{cb}^u, P_{lf}^u).

4.1 Aggregation

Aggregation gathers the individual preferences computed by the BRT modules for every member in the group G, and creates a single set of group preferences for each type of recommendation (P_d^G, P_{cb}^G,P_{lf}^G):

$$P_{brt}^G = \{(G, f, r^G) : \exists (u, f, r) \in \bigcup_{\forall u \in G} P_{brt}^u\}, \text{where } r^G = avg(r)$$

P_{brt}^G is the result of aggregating the preferences returned by the BRT for at least one user in the group. The interest-degree of a group preference r^G is calculated as the ***average value*** of the interest-degree of the users in G for the feature f.

The three lists of group preferences (P_d^G, P_{cb}^G and P_{lf}^G) are then used to obtain three lists of items to recommend. An item described under a feature f is included in a list if there is a tuple (G, f, r^G) that belongs to the corresponding group preference list. The three lists of items are combined by a **Hybrid RS**, which applies a mixed hybrid recommendation [3]. By handling these lists of items independently, we give much more flexibility to the GRSK because any other hybrid technique could be used instead.

The Hybrid RS returns a single list of ranked items (RC^G) whose elements are tuples of the form (i, Pr^i), where $i \in I$ is an item to recommend, and Pr^i is the estimated interest-degree of the group in the item i. This latter value is calculated as follows:

$$Pr^i = percentile(AC^i) + avg_{\forall f}(r^i + r^G) \tag{1}$$

where $percentile(AC^i)$ refers to the percentile rank of the acceptance counter of i (AC^i) with respect to the whole set of items accepted by the users when recommended. The second part of the formula considers the average interest-degree in all the features that describe the item i in both the taxonomy (r^i) and in the group preferences (r^G). The hybrid RS finally selects the best ranked items as the final group recommendations RC^G.

4.2 Intersection

The intersection mechanism finds the preferences that are shared by all the members in the group and make up the group preferences.

$$P_{brt}^G = \{(G, f, r^G) : \exists (u, f, r) \in \bigcap_{\forall u \in G} P_{brt}^u\}, \text{where } r^G = avg(r)$$

Preferences	D	CB	LF
User 1	(NS,90), (W,50)	(P,30)	
User 2	(NS,70)		(W,70)
User 3	(NS,80), (MM,100)	(P,50)	

NS: Nautical Sport P: Park
W: Walk MM: Modern Museum

Preferences	D	CB	LF
Intersection	(NS,80)		
Aggregation	(NS,80), (W,50), (MM,100)	(P, 40)	(w, 70)

Items Intersection		Items Aggregation	
Valencia Port	210	Ivam	230
		Valencia Port	200
		Botanical Garden	170

Fig. 4. Example of group recommendation (Aggregation and Intersection).

The final list of recommended items RC^G is computed as above from the three lists P_d^G, P_{cb}^G and P_{lf}^G.

Figure 4 shows an example of the recommendation process when using the aggregation and intersection mechanisms. This example is based on the taxonomy in figure 1. The table on the left shows the lists of preferences computed by each BRT. The intersection method obtains only one preference (*Nautical Sport*) because it is the only feature shared by all the group members. On the other hand, the aggregation method creates one list per BRT with the individual preferences of all the users. For example, the r^G value associated with *Nautical Sport* is computed as the average of the r^u values of all the group members.

When using the intersection, the system will recommend only items described under the feature *Nautical Sport*; in the taxonomy of figure 1, only one item is associated to this feature, *Visit to Valencia Port*. Assuming that $percentile(AC_{ValenciaPort})$ is 50, the priority of this item is computed as (r^i and r^G are 80): $Pr_{ValenciaPort} = 50 + avg(80 + 80) = 210$. On the other hand, when using the aggregation, all the items can be recommended; the final recommendations will depend on the priority of each item. For example, in this case, the priority of *Visit to Valencia Port* is computed as: $Pr_{ValenciaPort} = 50 + avg(80 + 80, 90 + 50) = 200$; this item is described by the features *Nautical Sport* and *Walk* with r^i values of 80 and 90, respectively. The first three items in the list will be recommended, as the group has requested three recommendations. It is important to note that the *IVAM Museum* will be recommended although only one user has *modern museum* among his preferences.

4.3 Incremental Intersection

The preferences predicted for the group are some function of all of the known preferences for every user in the group. Social Choice theorists, concerned with the properties of voting methods, have been investigating preference aggregation for decades. A very popular work is that of Arrow [2] which demonstrates the impossibility of combining individual preferences into a single expression of social preference in a way that satisfies several desirable properties. However, there are other investigations, specifically on Collaborative Filtering RS, that show that the only possible form for the prediction function is a weighted average of the users' ratings. The Incremental Intersection (II) method is actually a weighted average of the most voted preferences among the users in the group, that is the preferences shared by the largest possible group of members.

Preferences	D	CB	LF
User 1	(NS,90), (W,50)	(P,30)	
User 2	(NS,70)		(W,70)
User 3	(NS,80), (MM,100)	(P,50)	

Preferences II	
(NS,80)	3 votes
(W,60), (P,40)	2 votes
(MM,100)	1 vote

Items recommended	
Valencia Port	205
Botanical Garden	170
Cabecera Garden	150

Fig. 5. Example of group recommendation (Incremental Intersection).

The II method draws up a joint list for each user with the preferences computed by all the BRT. If more than one BRT returns a preference for the same feature f, the II builds a single preference (u, f, r^u) where r^u is the average interest-degree of all the preferences for f. The II method starts a voting process where a feature f is voted by a user if there is a preference for the feature f in his joint list.

The items to recommend will be the ones that describe the most voted features. At the first iteration, we select the features with the highest number of votes ($|G|$ votes), where the value r^G associated with each feature is computed as the average of r^u for all the users with preferences with f. Then, the items that describe these features are selected and their Pr^i is computed as equation 1 shows. If there are not enough items to cover the requested number of recommendations, at the next iteration we select the features that received at least $|G|-1$ votes, and so on. This way, we incrementally consider the features shared by the largest possible number of users in the group.

Figure 5 shows an example of the recommendation process when using the II. The preferences computed by each BRT for the group members are the same as in figure 4. In this case, we obtain several lists of preferences ordered by the number of votes of the features contained in the list. In the first iteration, only one item associated to the most-voted feature is recommended, namely *Visit the Valencia Port*. As the group has requested three recommendations, a second iteration will consider the features with at least two votes. In this case, three items are recommended, which are shown in figure 5 together with their calculated priority. It is important to remark that, unlike the aggregation method, the II does not recommend *IVAM Museum* at the second iteration because the feature that describes this item will only appear in the third iteration.

5 Experimental results

This section shows the experimental results performed to compare the three methods for the elicitation of the group preferences. As we are working with our own domain, our first task was to obtain data from real users. We prepared a questionnaire with questions about general preferences, demographic data, visited places and the user's degree of satisfaction when visiting the places. The questionnaire was filled in by 60 people and these data were used to create several groups with a different number of users.

Unlike individual recommendations, when dealing with groups, the most important issue is that the recommendation is as satisfactory as possible for all the members in the group. Thus, through the experimental setup presented in this section, our intention is to analyse which of the three techniques described above obtains the best recommendations as for the whole group satisfaction.

Let RC^u be the recommendation for a single user u, such that each element in RC^u has the form (i, u, Pr_u^i), where i is the recommended item $(i \in I)$, u is the user and Pr_u^i is the priority of the item i for the user u. Pr_u^i is set equal to 0 if this value is unknown for a given item i. Given a recommendation RC^G for a group G, such that $u \in G$, the **utility** of the user u with respect to RC^G is calculated as:

$$U_u^G = \sum_{\forall i \in RC^G} Pr_u^i$$

Thus, in order to analyse the quality of the recommendations, we consider the average and the standard deviation (dispersion) of the utility over all the group members: $\mu_G(U_u^G)$ and $\sigma_G(U_u^G), \forall u \in G$.

We executed three experiments, one per each preference elicitation mechanism, namely aggregation (Aggr), intersection (Int) and incremental intersection (II). We used groups of different size ranging from 2 to 6 members; the number of requested recommendations is set to 10 items in all cases. We also run the three experiments twice: the first batch is run with user ratings on the 30% of the items (Fig. 6 Exp-1-a), and the second one with user ratings on the 70% of the items (Fig. 6 Exp1-b). The X axis indicates the number of members in the group. Bars represent the utility on average of the recommendations obtained for each group size and elicitation mechanism. Likewise, the points in the lines determine the dispersion level for each group size and elicitation mechanism. Notice that bars are referred to the scale on the left whereas lines refer to the scale on the right. It can be observed that, in both sets of experiments, and for every group size, the utility on average is quite similar in all cases, whereas the dispersion (standard deviation) is lower in the incremental intersection than in the aggregation technique. The reason behind is that the II incrementally considers the preferences that satisfy a larger number of users whereas the aggregation recommends the most prioritized items for *at least one* member in the group, which obviously does not imply to be for *all* the group members. Therefore, we can conclude that the II obtains solutions of a similar utility as the aggregation technique but with a lower degree of dispersion, which is interpreted as all members in the group are equally satisfied.

On the other hand, in the first set of experiments (Fig. 6 Exp-1-a), the results of the II and the Int technique coincide (so we do not include the results of the Int technique), because the first intersection computed by II is enough to obtain the number of requested recommendations. However, in the second set of experiments, when the user has rated a larger number of items, it can be the case that the result of the Int mechanism does not cover the number of requested recommendations. A more detailed representation is shown in Figure 7, which compares the number of recommendations obtained by the II and the Int techniques in the second set of experiments for groups of 4, 5 and 6 members.

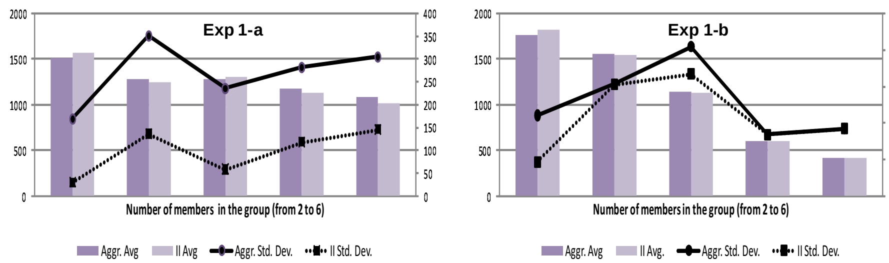

Fig. 6. Experiment 1: Comparison of the quality of the recommendations (II and Aggr)

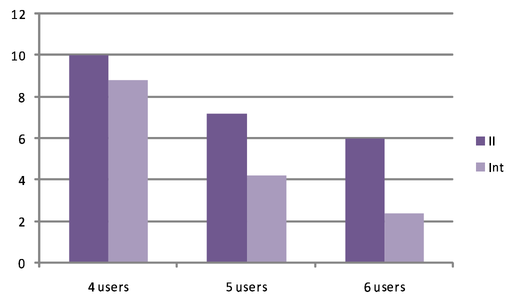

Fig. 7. Comparison of the number of recommendations (II and Int).

It can be observed that as the number of members increases, the Int mechanism finds more difficult to return a large set of recommendations. It is in these cases when the usefulness of the II mechanism shows up, because the incremental consideration of smaller groups helps to find more suitable items to recommend. In groups of 4 members, a large number of recommendations satisfy all members in the group; in groups of 5 members and 6 members, as more users contribute to the final recommendation, the number of recommendations that satisfy all the members lessens and so the results of II and aggregation become much more similar (this effect can also be observed in Fig. 6 Exp-1-b).

We can conclude that the II mechanism obtains the best results, because it brings together the benefits of the Aggr and the Int techniques.

6 Related work

Systems for recommending items to a group of two or more users in a tourist domain are particularly useful as people usually make group travels (family, friends, etc.). We will illustrate some group recommender systems for tourism such as *Intrigue*, *Travel Decision Forum* or *CATS*.

Intrigue [1] assists a group of users in the organization of a tour. Individual participants are not described one by one but the system models the group as a set partitioned into a number of homogeneous subgroups, and their possibly conflicting individual preferences are separately represented. *Intrigue* uses socio-demographic information about the participants and it elicits a set of preferences to define the subgroup requirements on the properties of tourist attractions, paying attention to those preferences possibly conflicting between subgroups. The group information stores a relevance value to estimate the weight that the preferences of a member should have on the recommendation. Each subgroup may have a different degree of influence on the estimation of the group preferences.

CATS [7] is a conversational critique-based recommender that helps a group of members to plan a skiing vacation. The recommender manages personal as well as group profiles. The individual preferences are elicited by subsequently presenting a recommendation to the user. By critiquing a recommendation, the user can express a preference over a specific feature in line with their own personal requirements. The group profile is maintained by combining the individual user models and associating critiques with the users who contributed them. The group recommendation displays information relating to group compatibility.

The *Travel Decision Forum* [6, 5] uses animated characters to help the members of a group to agree on the organization of a vacation. At any given moment, at most one member is interacting with the system, the other users in the group are represented as animated characters. The system uses a character that represents a mediator, who directs the interaction between the users. Users must reach an agreement on the set of preferences (group profile) that the recommendation must fulfil. Initially, each user fills out a preference form associating a degree of interest to each preference. The individual user profile as well as the group profile contain the same set of preferences. The degree of interest of a specific group profile is calculated out of the degree of interest of each user in the group. Once the group profile has been created, all group members must agree on the group preference model. The mediator asks each member of the group in turn whether the group model can be accepted or not. Using the users critiques, the mediator reconfigures the preferences ratios, and the recommendation is done using the group preference model.

All the described recommender systems only use aggregation methods to compute the group profile. In contrast, our approach uses three different mechanisms: aggregation, intersection and incremental intersection. Another distinguish characteristic is that *e-Tourism*, instead of making recommendations that directly match the group preferences (likes), it applies a more sophisticated technique: a hybrid recommendation technique that combines demographic, content-based recommendation and preference-based filtering.

7 Conclusions and Further Work

e-Tourism is a web-based service to make recommendations about personalized tourist tours in the city of Valencia (Spain) for a group of users. The component in charge of the recommendation process is the GRSK, a taxonomy-driven domain-independent recommendation kernel. Group recommendation are elicited from the individual preferences of each single user by using three basic recommendation techniques: demographic, content-based and preference-based filtering. For each recommendation technique, we compute a list of group preferences through the application of aggregation, intersection and incremental intersection methods. Finally, constrains yield to a list of items to recommend. The evaluation of this process shows that the incremental intersection is able to work in a great variety of situations, because it brings together the benefits of the aggregation and intersection techniques.

The design of the GRSK allows us to easily add new basic, hybrid or group recommendation techniques. This is an important contribution to be able to check and measure the effectiveness of a different technique for the domain where GRSK is being applied. We are actually developing a feedback process to adapt the GRSK behaviour through consecutive interactions between the group and the system.

Finally, we are also working in the use of agreement techniques to obtain group recommendation. The members of the group are modeled as agents who attempt achieving a reconciled solution for the whole group maximizing the user satisfaction. This technique allows us to include more sophisticated user behaviours into the group.

Acknowledgments. This work has been partially funded by Consolider Ingenio 2010 CSD2007-00022, Spanish Government Project MICINN TIN2008-6701-C03-01 and Valencian Government Project Prometeo 2008/051.

References

1. Ardissono L., Goy A., Petrone G., Segnan M., Torasso P. Intrigue: personalized recommendation of tourist attractions for desktop and handset devices. *Applied AI, Special Issue on Artificial Intelligence for Cultural Heritage and Digital Libraries*, 17(8-9):687–714, 2003.
2. Arrow K.J. *Social Choice and Individual Values*. Yale University Press, 1963.
3. Burke R. *The Adaptive Web*, chapter Hybrid web recommender systems, pages 377–408. Springer Berlin / Heidelberg, 2007.
4. Hanani U., Shapira B., Shoval P. Information filtering: Overview of issues, research and systems. *User Modeling and User-Adapted Interaction*, 11, 2001.
5. Jameson A. More than the sum of its members: Challenges for group recommender systems. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, 2004.
6. Jameson A., Baldes S., Kleinbauer T. Enhancing mutual awareness in group recommender systems. In Bamshad Mobasher and editors Sarabjot S. Anand, editors, *Proceedings of the IJCAI*, 2003.
7. McCarthy K., McGinty L., Smyth B., Salam M. Social interaction in the cats group recommender. In *Workshop on the Social Navigation and Community based Adaptation Technologies*, 2006.
8. Sebastia L., Garcia I., Onaindia E., Guzman C. e-Tourism: a tourist recommendation and planning application. In *20th IEEE International conference on Tools with Artificial Intelligence (ICTAI)*, 2008.