Groups Identification and Individual Recommendations in Group Recommendation Algorithms [*] [†]

Ludovico Boratto
Dipartimento di Matematica e
Informatica, Università di
Cagliari
Via Ospedale 72
09124 Cagliari, Italy
ludovico.boratto@unica.it

Salvatore Carta
Dipartimento di Matematica e
Informatica, Università di
Cagliari
Via Ospedale 72
09124 Cagliari, Italy
salvatore@unica.it

Michele Satta
Dipartimento di Matematica e
Informatica, Università di
Cagliari
Via Ospedale 72
09124 Cagliari, Italy
michele_satta@hotmail.com

ABSTRACT
Recommender systems usually deal with preferences previously expressed by users, in order to predict new ratings and recommend items. To support recommendation in social activities, group recommender systems were developed. Group recommender systems usually consider predefined/a priori known groups and just a few existing approaches are able to automatically identify groups.

When groups are not already formed, another key aspect of group recommendation is related to groups identification. In this paper a novel algorithm able to identify groups of users and produce recommendations for each group is presented. The algorithm uses individual recommendations and a classic clustering algorithm to identify and model groups. Experimental results show how this approach substantially improves the quality of group recommendations with respect to the state-of-the-art.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: Information filtering; H.4 [**Information Systems Applications**]: Miscellaneous; M.4 [**Knowledge Modeling**]: Miscellaneous

General Terms
Algorithms, Experimentation, Performance

Keywords

Group Recommendation, Collaborative Filtering, Clustering

1. INTRODUCTION
With the development of Web 2.0, the use of the web has become increasingly widespread and users have had the chance to express opinions about shared content updated daily. This generates an incredible amount of data that can't be handled directly by the users. So finding relevant information over the Internet nowadays is becoming more and more difficult [18].

Recommender systems have been developed to deal with information overload and produce personalized content for the users by exploiting context-awareness in a domain. This is done by computing a set of previously expressed preferences, in order to recommend items that are likely of interest to a user. Collaborative Filtering (CF) [11, 15, 19] is by far the most successful recommendation technique. The main idea of CF systems is to use the opinions of a community, in order to provide item recommendations.

There are context and domains where classic recommendation cannot be used, because the recommendation process involves more than a person and preferences have to be combined in order to produce a single recommendation that satisfies everyone (e.g., people traveling together or going to a restaurant/museum together). Therefore, in order to support recommendations in social activities, algorithms able to provide group recommendations were developed. Group recommendations are provided according to the way a group is modeled. Group modeling is the combination of the preferences expressed by single users into a common group preference.

A special type of group recommendation is needed when technological constraints limit the bandwidth available for the recommendation. This is for example the case of Satellite Systems, in which the number of channels is limited and a personalized TV schedule cannot be produced.

Another useful application scenario in which limitations are imposed in the recommendation process is the printing of recommendation flyers that contain suggested items. Even if a company has all the data to produce a flyer with individual recommendations for each customer, the process of printing

a different flyer for everyone would be technically too hard to achieve and costs would be too high. A possible solution would be to print n different flyers that can be affordable in terms of costs and that can satisfy users by recommending interesting items to the recipients of the same flyer.

In both the scenarios described the first result that the algorithm has to compute is a proper identification of groups, in order to produce a recommendation that maximizes users satisfaction. This preliminary phase of the group recommendation process is not performed by the great part of algorithms in literature, because they consider only how to model already existing groups.

In this paper a novel approach for group recommendation with automatic identification of groups is proposed.

To enhance the readability of the paper and the properties of the proposed approach, a baseline version of the algorithm is preliminarily presented (*BaseGRA*, Baseline Group Recommendation Algorithm). *BaseGRA* uses a classic clustering algorithm to identify groups, by exploiting past preferences expressed by each user of the system. To model the group, *BaseGRA* combines the preferences of each user with the ratings predicted using a CF algorithm for the unrated items.

Since the number of items evaluated by a user in a system is usually much lower than the number of the items that can be evaluated, we considered the fact that the clustering step may be affected by the well-known problem of *sparsity* of the available data.

The algorithm presented in this paper, named *Improved-GRA* (Improved Group Recommendation Algorithm), has been developed to overcome this potential problem and improve the quality of clustering. This is done by using the predictions of the missing ratings to complete the matrix of the preferences already expressed by users. The algorithm predicts individual recommendations, combines them with the preferences explicitly expressed by users, and uses both of them as input for a classic clustering algorithm. As highlighted by the experiments, this leads to an identification of groups of users with similar preferences with a high quality of the predicted results. Individual recommendations and explicitly provided preferences are also used to model the groups.

The proposed approach is the first that combines clustering of the users with an aggregation of individual recommendations. In fact none of the existing recommender systems that automatically identify groups merges individual recommendations and the approaches that merge individual recommendations deal with groups that have a predefined structure.

Another scientific contribution of the approach relies in the algorithm used to automatically identify groups, which mixes recommendation and clustering algorithms, leading to a substantial improvement of the quality of the group recommendations with respect to the state-of-the-art.

Moreover the paper presents an analysis of two more funda-

mental aspects of this kind of group recommendation: homogeneity of group size and homogeneity of recommendations quality.

Considering the size of groups, it is evident that it should be sufficiently homogeneous. In simple words, if the recommendation process involves 70000 users and 10 available channels, it would not be acceptable to have a group with 61000 users and 9 groups with 1000 users. In fact it would be a waste of bandwidth to produce recommendations for small groups and, at the same time, it would be hard for a system to produce recommendations that gather the preferences of a large group.

Considering the quality of the predicted results, it should not vary too much between the groups. In other terms, the system should try to keep a sufficient quality of the predictions for every group. Providing inadequate recommendations to any group should always be avoided.

The rest of the paper is organized in the following way: section 2 presents related work, considering both group recommender systems able to automatically identify groups and group recommender systems that build individual recommendations; section 3 contains a detailed description of the baseline group recommendation algorithm, BaseGRA; section 4 will do the same for the improved algorithm Improved-GRA; section 5 describes the experiments we conducted to evaluate the proposed algorithm and outlines main results; section 6 contains comments, conclusions and future developments.

2. RELATED WORK

As mentioned in the Introduction, group recommender systems were developed to support the recommendation process in activities that involve more than a person.

In [13] and [5] the state-of-the-art in group recommendation is presented. The existing systems were developed for different domains like web/news pages, tourist attractions, music tracks, television programs and movies. A classification of those approaches can be made from two perspectives:

- the type of group considered;
- the way group recommendations are built.

Considering the first classification of the existing systems, which is based on the type of groups considered, we can identify four different types of groups, described below.

- **Established group**: a number of persons who explicitly choose to be part of a group, because of shared, long-term *interests*;
- **Occasional group**: a number of persons who do something occasionally together, like visiting a museum. Its members have a common *aim* in a particular moment;
- **Random group**: a number of persons who share an environment in a particular moment, without explicit interests that link them;
- **Automatically identified group**: groups that are automatically detected considering the preferences of the users and/or the resources available.

The second classification of the existing approaches can be done considering the way group recommendations are built. There are two ways to build group recommendations, described in the list below.

- Merge of individual recommendations into a group recommendation.
- Merge of the individual preferences to build a group profile and predict specific recommendations for the group.

The approach described in this paper automatically identifies groups and merges individual recommendations. The existing approaches for those two categories of group recommender systems will now be described and differences with our approach will be highlighted.

As a general consideration, please note that none of the approaches that automatically identify groups merges individual recommendations.

2.1 Approaches that automatically identify groups

The approach proposed in [8] aims to automatically discover Communities of Interest (CoI) (i.e., a group of individuals who share and exchange ideas about a given interest) and produce recommendations for them.

CoI are identified considering the preferences expressed by users in personal ontology-based profiles. Each profile measures the interest of a user in concepts of the ontology. Users interest is exploited in order to cluster the concepts.

User profiles are then split into subsets of interests, to link the preferences of each user with a specific cluster of concepts. Hence it is possible to define relations among users at different levels, obtaining a multi-layered interest network that allows to find multiple CoI. Recommendations are built using a content-based CF approach.

The difference with our approach is that preferences of users are not expressed through an ontology. Moreover, our recommendation technique is based on a CF user-based approach.

The system proposed in [6] generates group recommendations and automatically detects intrinsic communities of users whose preferences are similar. Communities of users with similar preferences are identified using a Modularity-based Community Detection algorithm [4] and group recommendations are predicted for each community. See 5.2 for a more detailed description of the approach.

This approach, although it achieves exactly the same purposes, differs from the one presented in this paper both in the way group predictions are built and in the way groups are identified. The approach was chosen for comparison with the algorithm presented in this paper because of the mentioned similarities in several aspects.

2.2 Approaches that merge individual recommendations

PolyLens [17] is a system built to produce recommendations for groups of users who want to see a movie.

To produce recommendations for each user of the group a CF algorithm is used. In order to model the group, a "least misery" (LM) strategy is used: the rating used to recommended a movie to a group is the lowest predicted rating for that movie, to ensure that every member is satisfied.

In contrast with the LM strategy used by PolyLens, in our approach group preferences are built combining individual recommendations in a single value that averages the preferences of the single users.

We considered the use of a group modeling technique based on the average of users ratings instead of using a LM strategy because it seems more suited for an approach where large groups are considered. A LM strategy is useful for small groups and in fact Polylens handles groups with two or three users. Even if groups are composed by people with homogeneous preferences, using a LM strategy a low rating expressed by a user for a movie would be enough to have a low rating for that movie for the whole group. With large groups such an approach would probably lead to extremely low ratings for almost all the movies.

INTRIGUE (INteractive TouRist Information GUidE) [2, 3] is a system that recommends sightseeing destinations using the preferences of the group members. The approach merges individual recommendations and, in order to build group recommendations, some subgroups are considered more influential (e.g., disabled people).

In our approach we don't consider a specific domain of application and every individual recommendation is weighted equally, so that group recommendations reflect all the users preferences.

The approach presented in [1] computes group recommendations by combining individual recommendations built for every user and considering a *consensus function*, which combines *relevance* of the items for a user and *disagreement* between members.

Since our approach automatically builds groups of users with similar preferences, we don't expect disagreement to be a characterizing feature when computing group recommendations. Therefore this aspect was not considered in our approach.

The system proposed in [9, 10] presents a group recommendation approach based on Bayesian Networks (BN). To represent users and their preferences a BN is built. The authors assume that the composition of the groups is a priori known and model the group as a new node in the network that has the group members as parents. A collaborative recommender system is used to predict the votes of the group members. A posteriori probabilities are calculated to combine the predicted votes and build the group recommendation.

The main difference with our approach is that, in order to combine preferences and build group recommendations, we

don't rely on a Bayesian Network and a posteriori probabilities.

3. BASELINE GROUP RECOMMENDATION ALGORITHM (BASEGRA)

The baseline version of our algorithm identifies groups of similar users considering the preferences expressed by each user and models each group using individual recommendations built for each user of a group.

3.1 Overview of BaseGRA

The algorithm works in two steps:

1. Using a *Ratings Matrix* that contains the preferences of each user, groups of similar users are detected through the k-means clustering algorithm [14].

2. Once the groups have been detected, a group preference is produced by aggregating the preferences of the individual users.

3.2 Groups Identification

The input of the algorithm is a *Ratings Matrix M* that associates a set of *users* to a set of *items* through a *rating*. A rating indicates the level of satisfaction of a user for a considered item. So each value m_{ui} of the Ratings Matrix is:

$$m_{ui} = \begin{cases} r_{ui} & \text{if user } u \text{ expressed a preference for item } i \\ \emptyset & \text{if user } u \text{ didn't express a preference for item } i \end{cases}$$

A rating r_{ui} is always such that $r_{min} \leq r_{ui} \leq r_{max}$ and $r_{ui} > 0$. In other words, a rating value is always inside a fixed range and its value is always positive.

The Ratings Matrix is used as input for the k-means clustering algorithm [14]. Since the algorithm's input are the preferences expressed by each user, the output is a partition in groups of users with similar preferences.

3.3 Groups Modeling

The objective of group modeling is to calculate, for each item, a group rating which will be evaluated in order to decide which items should be recommended to the group. In order to model a group, the preferences of each user that belongs to the group have to be combined.

An average is a single value that is meant to typify a list of values. The most common method to calculate such a value is the arithmetic mean, which also seems an effective way to put together the preferences of each user in a group, in order to reach our objective.

Combining just the preferences expressed by the users would lead to a poor modeling of the group, since each user usually gives an explicit preference to a small set of item. This is especially true when modeling small groups. In fact group preferences have to be extracted considering a small set of preferences expressed by a small set of users.

In order to improve the efficiency of group modeling, our algorithm completes the Ratings Matrix, adding individual

recommendations predicted for each user. The result is a *Predicted Ratings Matrix PR* that associates each user u with an item i either through an explicitly expressed rating r_{ui} or through a predicted rating p_{ui}.

A predicted rating p_{ui} is calculated using a classic User-Based Nearest Neighbor CF Algorithm, proposed in [20]. The algorithm predicts a rating p_{ui} for each item i that was not evaluated by a user u, considering the rating r_{ni} of each similar user n for the item i. A user n similar to u is called a *neighbor* of u. Equation 1 gives the formula used to predict the ratings:

$$p_{ui} = \overline{r}_u + \frac{\sum_{n \subset neighbors(u)} sim(u,n) \cdot (r_{ni} - \overline{r}_n)}{\sum_{n \subset neighbors(u)} sim(u,n)} \quad (1)$$

Values $\overline{r}_u$ and $\overline{r}_n$ represent, respectively, the mean of the ratings expressed by user u and user n. Similarity $sim()$ between two users is calculated using the Pearson correlation, a coefficient that compares the ratings of all the items rated by both the target user and the neighbor (*corated* items). Pearson correlation between a user u and a neighbor n is given in Equation 2. $CR_{u,n}$ is the set of corated items between u and n.

$$sim(u,n) = \frac{\sum_{i \subset CR_{u,n}} (r_{ui} - \overline{r}_u)(r_{ni} - \overline{r}_n)}{\sqrt{\sum_{i \subset CR_{u,n}} (r_{ui} - \overline{r}_u)^2} \sqrt{\sum_{i \subset CR_{u,n}} (r_{ni} - \overline{r}_n)^2}} \quad (2)$$

4. IMPROVED GROUP RECOMMENDATION ALGORITHM (IMPROVEDGRA)

BaseGRA identifies groups of similar users using a Ratings Matrix, i.e., a matrix that contains all the preferences expressed by users for the evaluated items.

However, the number of items rated by users is much lower than the number of available items. This leads to the sparsity problem that is common in clustering.

ImprovedGRA was conceived to improve the quality of the clustering step of BaseGRA. ImprovedGRA identifies groups giving as input to the k-means algorithm not the original Ratings Matrix M, that contains the ratings already expressed by users, but the complete *Predicted Ratings Matrix PR* previously presented, where the predicted values of the unrated items for each user are added.

In order to do so, the individual recommendations are predicted by ImprovedGRA at the beginning of the computation. Using more values as input for the clustering, the algorithm should be able to identify better groups, i.e., groups composed by users having more correlated preferences. This should lead to a higher overall quality of the group recommendations.

In conclusion, ImprovedGRA performs the same steps performed by BaseGRA but computes individual recommen-

dations before clustering the users. This allows to cluster the users using more preferences and identify better groups. The preferences expressed by users and the individual recommendations are also used to model the group.

5. EXPERIMENTS

In this section we first describe the strategy and aims which drove our experiments.

Then a state-of-the-art group recommender system that automatically identifies groups, chosen for comparison with the proposed approach, is described.

Experiments setup and metrics used are then described and, at the end of the section, results are shown and commented.

5.1 Experimental Methodology

In order to evaluate the quality of the system, three aspects were considered: quality of the predicted ratings, distribution of the quality between the groups and homogeneity of the groups size. The details of each experiment will be described next.

5.1.1 Quality of the predicted ratings evaluation

The main objective of a recommender system is to produce high quality predictions. The algorithm presented in this paper produces group recommendations adapting to the bandwidth available for the recommendation process.

In order to evaluate the quality of the predicted ratings for different bandwidths, i.e., for different numbers of channels that can be dedicated to the recommendation, we built three different partitions of the users in groups. A partition is a set of n groups in which users are subdivided. Of course, if groups are homogeneous, the larger is n, the smaller are the groups and the system can predict better ratings, because the preferences of a small amount of users have to be combined.

In order to properly evaluate the performances of the proposed algorithms, we compared them with the results obtained considering a single group with all the users (predictions are calculated considering all the preferences expressed for an item), and the results obtained using no partition of the users (i.e., quality of the individual recommendations is calculated).

To measure the quality of the predicted ratings, we used the Root Mean Squared Error (RMSE). This metric was chosen because it is the most common in literature.

In order to analyze the quality of the predictions produced by each algorithm for different partitions, we produced a plot that shows the trend of RMSE for each partition in n groups.

5.1.2 Distribution of quality between the groups evaluation

A second important aspect that has to be evaluated is how the quality of the predicted results is distributed between the groups of a partition.

In fact a group recommender system should be able to distribute the quality of the predicted results in a sufficiently equal way, in order to satisfy the recommendation demand for all the users of the system.

To analyze how RMSE is distributed between the groups produced by ImprovedGRA, a table that contains the mean value of RMSE for each partition and how many groups have a RMSE value close/far to the mean is presented.

To compare the different algorithms, we measured the standard deviation of the RMSE values obtained for every group of a partition.

5.1.3 Distribution of size between the groups evaluation

The last aspect we evaluated is how homogeneous are the groups in terms of size. Indeed, it is not acceptable to have too large or too small groups. At the same time the clustering step cannot create an homogeneity which is not intrinsically existent in users. To evaluate this trade-off we measured the standard deviation of the size of the groups present in a partition.

5.2 Benchmark algorithm: ModularityBasedGRA

The technique selected for comparison with ImprovedGRA, is the one proposed in [6]. From now on, the algorithm will be called ModularityBasedGRA, because of the approach used to identify groups (based on the Modularity function).

ModularityBasedGRA is an algorithm that generates group recommendations and automatically detects intrinsic communities of users whose preferences are similar. The input is a Ratings Matrix that associates a set of *users* to a set of *items* through a *rating*. Based on the ratings expressed by each user, the algorithm evaluates the level of similarity between users and generates a network that contains the similarities.

A modularity-based Community Detection algorithm proposed in [4] is run on the network in order to find partitions of users in communities. For each community, ratings for all the items are predicted using an item-based CF algorithm.

Since the Community Detection algorithm is able to produce a dendrogram, i.e. a tree that contains hierarchical partitions of the users in communities of increasing granularity, the quality of the recommendations can be evaluated for the different partitions.

To achieve the objectives previously outlined, i.e., detect the communities and produce group recommendations for them, ModularityBasedGRA computes four steps, described below.

Users similarity evaluation In order to create communities of users, the algorithm takes as input a *Ratings Matrix* and evaluates through a standard metric (cosine similarity) how similar the preferences of two users are. The result is a weighted network where nodes rep-

resent users and each weighted edge represents the similarity value of the users it connects. A post-processing technique is then introduced to remove noise from the network and reduce its complexity.

Communities detection In order to identify intrinsic communities of users, a Community Detection algorithm proposed by [4] is applied to the users similarity network and partitions of different granularities are generated.

Ratings prediction for the items rated by the group A group's ratings are evaluated by calculating, for each item, the mean of the ratings expressed by the users of the group. In order to predict meaningful ratings, the algorithm calculates a rating only if an item was evaluated by a minimum percentage of users in the group. With this step it is not possible to predict a rating for each item, so another step was created to predict the remaining ratings.

Ratings prediction for the remaining items For some of the items, ratings could not be calculated by the previous step. In order to estimate such ratings, similarity between items is evaluated, and the rating of an item is predicted with a CF item-based algorithm that considers the items most similar to it.

The choice to compare ImprovedGRA with this approach is motivated by the fact that both approaches produce group recommendations and automatically identify groups of users. Moreover, both can be evaluated for different partitions of users in groups. This allows a direct comparison between the two approaches.

Let us also note that even if the aim of the two algorithms is the same, the two techniques work in completely different ways: ImprovedGRA clusters users with a classic algorithm (k-means) after building individual recommendations and then models the groups preferences, while ModularityBasedGRA clusters users with a Community Detection algorithm and then builds group recommendations.

5.3 Experiments Setup

The experimentation was made using the MovieLens-1M dataset, which is composed of 1 million ratings, expressed by 6040 users for 3900 movies. In order to evaluate the quality of the ratings predicted by each of the algorithms, around 20% of the ratings was extracted as a test set and the rest of the dataset was used as a training set for the algorithm.

Each group recommendation algorithm was run with the training set and, for each partition of the users in groups, ratings were predicted.

The obtained values were used to conduct the experiments previously described.

5.4 Evaluation metrics

This section will introduce the two metrics used to evaluate different characteristics of our algorithm, the Root Mean Squared Error (RMSE) and the Standard deviation. Both metrics compare the obtained results with a comparison value, in order to evaluate the quality of the system.

5.4.1 Root Mean Squared Error (RMSE)

The quality of the predicted ratings was measured through the Root Mean Squared Error (RMSE). The metric compares the test set with the predicted ratings: each rating r_{ui} expressed by a user u for an item i is compared with the rating p_{gi} predicted for the item i for the group in which user u is. The formula is shown below:

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n}(r_{ui} - p_{gi})^2}{n}}$$

where n is the number of ratings available in the test set.

5.4.2 Standard deviation

The homogeneity of the groups size and the distribution of RMSE between the groups was measured with the standard deviation (considering respectively the size of the groups and the RMSE values of the groups).

The metric evaluates how much variation there is from the "average" value. A low standard deviation indicates that the size of the groups/the RMSE obtained for the groups tend to be close to the mean, while high values of standard deviation indicate that the obtained values are scattered over a large range of values.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

5.5 Experimental results

The first experiment, presented in 5.1.1, aims to evaluate the quality of the predicted values for a partition of the users in groups. Figure 1 shows the trend of the RMSE values for the different partitions of the users in groups.
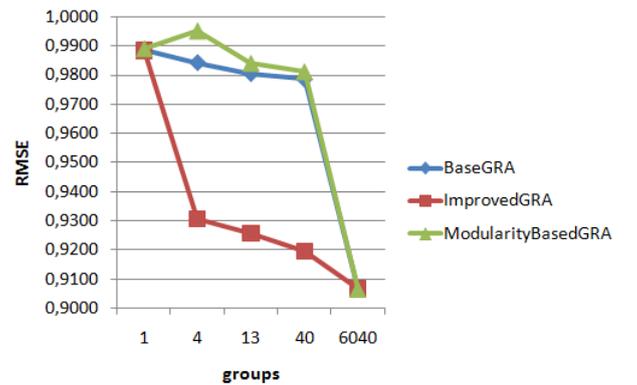
Figure 1: RMSE values for each partition

For all the algorithms, we can notice that as the number of groups grows, the quality of the recommendations improves, since groups get smaller and the algorithms can predict more precise ratings. We can see that the values of RMSE notably decrease when the algorithms start grouping the users (i.e., there is a big difference of RMSE between 1 and 4 groups). The RMSE values continue to decrease for the other partitions, but the improvement in quality is lower.

Comparing the algorithms, we can see that BaseGRA and ImprovedGRA outperform the benchmark algorithm ModularityBasedGRA. Moreover, the performances of Improved-GRA are much better than the performances of BaseGRA: this proves that enhancing the Ratings Matrix with individual recommendations leads to great improvements in the quality of the predicted results.

The second experiment, presented in 5.1.2, was conducted to evaluate how the quality of the predicted values is distributed between the groups. To do so we measured the standard deviation of RMSE of the groups in each partition.

Partition	Number of groups with RMSE r			
4 groups $\bar{x} = 0,93$	$r = 0,85$	$r = 0,89$	$r = 0,95$	$r = 1,04$
	1	1	1	1
13 groups $\bar{x} = 0,93$	$r < 0,87$	$0,87 \leq r \leq 1,00$		$r > 1,00$
	3	7		3
40 groups $\bar{x} = 0,96$	$r < 0,90$	$0,90 \leq r \leq 1,00$		$r > 1,00$
	15	15		10

Table 1: Distribution of RMSE between the groups

Table 1 shows, for each partition, the mean of the RMSE obtained for every group with ImprovedGRA and how the RMSE is distributed between the groups. As we can see, the majority of the groups in each partition has a RMSE value sufficiently close to the mean. This means that RMSE is distributed quite equally between the groups and our approach is able to satisfy the recommendation demand for all the users.
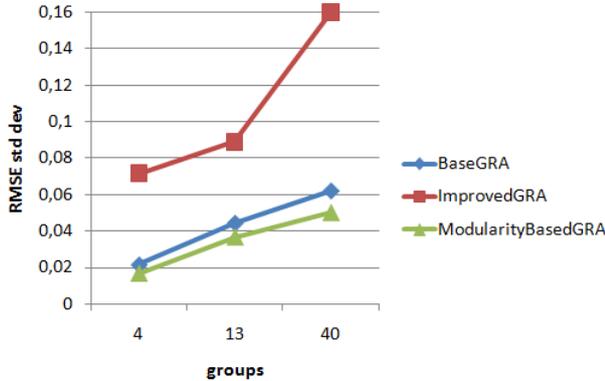
Figure 2: Standard deviation of RMSE of the groups

Figure 2 compares the standard deviation of RMSE of the groups for the different approaches. ImprovedGRA values are slightly higher if compared to the other approaches. However, it is important to remember that in this case there is a trade-off between an equal distribution in terms of RMSE and the similarity between the users in a group. In fact the groups have to be intrinsic in order to improve the quality of the predicted results. So it seems reasonable to loose a bit of homogeneity in distribution of the quality in order to improve the overall quality of the results predicted by the system. This is the case of ImprovedGRA in which the

RMSE is distributed less equally between the groups but the quality of the predictions compared with the other approaches is much higher.

The third experiment, presented in 5.1.3 was conducted to evaluate how the size of the groups is distributed in each partition (i.e., how homogeneous are the groups in terms of size). To do so we measured the standard deviation of the size of all the groups in each partition.

Partition	Number of groups with size s			
4 groups $\bar{x} = 1510$	$s = 633$	$s = 1334$	$s = 1807$	$s = 2266$
	1	1	1	1
13 groups $\bar{x} = 464, 62$	$s < 300$	$300 \leq s \leq 540$		$s > 540$
	3	7		3
40 groups $\bar{x} = 151$	$s < 80$	$80 \leq s \leq 250$		$s > 250$
	9	26		5

Table 2: Distribution of size of the groups

Table 2 shows, for each partition, the mean of the size obtained for every group with ImprovedGRA and how the size is distributed between the groups. As the table shows, most of the groups have size values close to the mean. This means that the size is distributed in a sufficiently equal way between the groups and our algorithm is able to produce recommendations properly, i.e., without handling the preferences of too small/large groups.
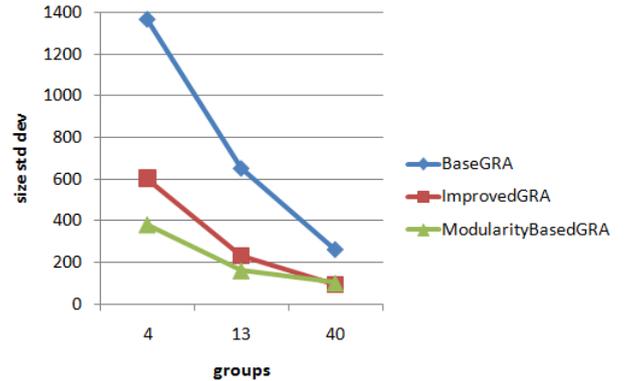
Figure 3: Standard deviation of size of the groups

Figure 3 compares the standard deviation of the size of the groups for the different approaches. It is important to notice how the enhancement of the Ratings Matrix made for ImprovedGRA leads to more homogeneous partitions in groups compared with BaseGRA.

The values obtained by ImprovedGRA are slightly higher than ModularityBasedGRA but also in this case there is a trade of between homogeneity of the groups size and similarity between the users. In fact it is important to find partitions of intrinsic groups with similar preferences that can lead to a high quality of the predicted results. So, a little loss in homogeneity of the size leads to great improvements in the quality of the results.

6. CONCLUSIONS AND FUTURE WORK

In this paper we presented an algorithm that combines user clustering with individual recommendations in order to identify and model groups of users with similar preferences and improve the quality of group recommendations in systems that automatically identify groups. In fact, BaseGRA and ImprovedGRA outperform the benchmark algorithm ModularityBasedGRA.

Moreover, we can notice that ImprovedGRA, using an enhanced Ratings Matrix to identify and model the groups, is able to produce sufficiently homogeneous groups in terms of size and distribution of RMSE. Therefore, all the three important objectives that should be achieved by a group recommender systems are reached by the proposed algorithm ImproveGRA.

Future developments of the algorithm have been planned for different steps performed by the algorithm. In [16] several strategies for group modeling were presented. We are currently studying how different strategies affect the quality of group recommendation with groups that are automatically identified.

Recently [7, 12] highlighted how different metrics to evaluate the quality of recommendation lead to completely different results. As a future work we plan to evaluate our systems with such metrics, in order to catch different aspects of our system.

7. REFERENCES

[1] S. Amer-Yahia, S. B. Roy, A. Chawla, G. Das, and C. Yu. Group recommendation: Semantics and efficiency. *PVLDB*, 2(1):754–765, 2009.

[2] L. Ardissono, A. Goy, G. Petrone, and M. Segnan. A multi-agent infrastructure for developing personalized web-based systems. *ACM Trans. Internet Technol.*, 5(1):47–69, 2005.

[3] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso. Intrigue: Personalized recommendation of tourist attractions for desktop and handset devices. *Applied Artificial Intelligence*, 17(8):687–714, 2003.

[4] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *J. Stat. Mech.*, 2008(10):P10008+, October 2008.

[5] L. Boratto and S. Carta. State-of-the-art in group recommendation and new approaches for automatic identification of groups. In G. A. Alessandro Soro, Eloisa Vargiu and G. Paddeu, editors, *Information Retrieval and Mining in Distributed Environments*. Springer Verlag. In press, 2010.

[6] L. Boratto, S. Carta, A. Chessa, M. Agelli, and M. L. Clemente. Group recommendation with automatic identification of users communities. In *Web Intelligence/IAT Workshops*, pages 547–550. IEEE, 2009.

[7] E. Campochiaro, R. Casatta, P. Cremonesi, and R. Turrin. Do metrics make recommender algorithms? *International Conference on Advanced Information Networking and Applications Workshops*, 0:648–653, 2009.

[8] I. Cantador, P. Castells, and E. P. Superior. Extracting multilayered semantic communities of interest from ontology-based user profiles: Application to group modelling and hybrid recommendations. In *Computers in Human Behavior, special issue on Advances of Knowledge Management and the Semantic.* Elsevier. In press, 2010.

[9] L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, and M. A. Rueda-Morales. Group recommending: A methodological approach based on bayesian networks. In *ICDE Workshops*, pages 835–844. IEEE Computer Society, 2007.

[10] L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, and M. A. Rueda-Morales. Managing uncertainty in group recommending processes. *User Model. User-Adapt. Interact.*, 19(3):207–242, 2009.

[11] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communication of the ACM*, 35(12):61–70, 1992.

[12] A. Gunawardana and G. Shani. A survey of accuracy evaluation metrics of recommendation tasks. *J. Mach. Learn. Res.*, 10:2935–2962, 2009.

[13] A. Jameson and B. Smyth. Recommendation to groups. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web: Methods and Strategies of Web Personalization.* Springer, 2007.

[14] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. L. Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[15] T. W. Malone, K. R. Grant, F. A. Turbak, S. A. Brobst, and M. D. Cohen. Intelligent information-sharing systems. *Communication of the ACM*, 30(5):390–402, 1987.

[16] J. Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Modeling and User-Adapted Interaction*, 14(1):37–85, 2004.

[17] M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl. Polylens: a recommender system for groups of users. In *ECSCW'01: Proceedings of the seventh conference on European Conference on Computer Supported Cooperative Work*, pages 199–218, Norwell, MA, USA, 2001. Kluwer Academic Publishers.

[18] S. Ram. Intelligent agents and the world wide web: Fact or fiction? *Journal of Database Management*, 12(1):46–49, 2001.

[19] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina, 1994. ACM.

[20] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*, volume 4321 of *Lecture Notes in Computer Science*, chapter 9, pages 291–324. Springer, 2007.