Modeling the Preferences of a Group of Users Detected by Clustering: a Group Recommendation Case-Study*

Ludovico Boratto, Salvatore Carta
Dipartimento di Matematica e Informatica
Università di Cagliari
Via Ospedale 72, 09124 - Cagliari, Italy
{ludovico.boratto, salvatore}@unica.it

ABSTRACT

Group modeling is the process that combines multiple user models into a single model. In group recommendation, this allows to derive a group preference for each item. It is known that the strategy used to model a group has to be chosen considering the domain in which the system operates. This paper evaluates group modeling strategies in a group recommendation scenario in which groups are detected by clustering users. Once users are clustered, strategies are tested, in order to find the one that allows to get the best accuracy. Experimental results show that clustering and group modeling are strongly connected. By producing group preferences that are equally distant from the individual preferences, the modeling strategy has the same role that the centroid has when users are clustered. This previously unknown link among the two tasks is essential in order to build accurate group recommendations.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—Clustering, Information filtering, Retrieval models, Selection process

General Terms

Algorithms, Experimentation, Performance

Keywords

Group Modeling, Group Recommendation, Clustering.

*This work is partially funded by Regione Sardegna under project SocialGlue, through PIA - Pacchetti Integrati di Agevolazione "Industria Artigianato e Servizi" (annualità 2010).

WIMS'14 June 2-4, 2014 Thessaloniki, Greece

ACM 978-1-4503-2538-7/14/06...$15.00.
http://dx.doi.org/10.1145/2611040.2611073 .

1. INTRODUCTION

Group recommendation [4] is designed for contexts in which more than one person is involved in the recommendation process [10]. In order to derive a group preference for the items, *group modeling* strategies combine the individual user models [14].

In [19], Pizzutilo et al. highlighted that "there is no strategy useful in every context independently from the environment". So, the strategy that best models a group has to be evaluated in the context in which the group is modeled.

A possible application scenario in which group recommendation can be used, is when the number of recommendations that can be produced is limited.

> *A company decides to print recommendation flyers that present suggested products. Even if the data to produce a flyer with individual recommendations for each customer is available, printing a different flyer for everyone would be technically too hard to accomplish and costs would be too high. A possible solution would be to set a number of different flyers to print, such that the printing process could be affordable in terms of costs and the recipients of the same flyer would be interested to its content.*

In order to respect the constraint on the number of recommendations that can be produced, this type of group recommender systems also has to detect groups.

This paper analyzes the problem of modeling the preferences of groups detected by clustering users. Since no group recommender system detects groups, this is the first time that group modeling is studied in this context.

The scientific contributions coming from this paper are now presented:

- by studying group modeling in a novel group recommendation context, this paper is the first to evaluate, compare and analyze the performance of group modeling strategies for groups detected by clustering users;
- our work shows how group modeling and clustering are linked and how a modeling strategy is influenced by the type of group it handles;
- by studying group modeling in a new context, this work shows *which* strategies can be used, *when* they can be used (under which circumstances), and what is the *best* strategy that models groups detected through clustering.

The rest of the paper is organized as follows: Section 2 presents the existing group modeling strategies, with references to the main works in the literature that use them; Section 3 describes the group recommender systems used for the evaluation; Section 4 presents the experiments conducted to evaluate the strategies and outlines main results; Section 5 contains conclusions and future work.

2. GROUP MODELING

This section illustrates the group modeling strategies presented in [14]. Each strategy is described, along with a reference to the group recommender systems that use it.

Each description contains an example of how individual ratings are combined by the strategy, by considering three users (u_1, u_2 and u_3) that rate ten items (identified with the letters from A to J) with a rating from 1 to 10.

2.1 Additive Utilitarian Strategy

Individual ratings for each item are summed and a list of the group ratings is produced (the higher the sum is, the earlier the item appears in the list). The ranked group list of items is exactly the same that would be produced when averaging the individual ratings, so this strategy is also called 'Average strategy'.

Pocket RestaurantFinder [15] recommends restaurants to a group of people, by averaging the individual preferences of the group members on different types of features (e.g., location, cost, cuisine). In [18], authors illustrate that modeling users with an average is the best way to model individual preferences in different contexts; so, this strategy is employed in their experiments, along with the Average Without Misery strategy, which will be presented later.

	A	B	C	D	E	F	G	H	I	J
u_1	8	10	7	10	9	8	10	6	3	6
u_2	7	10	6	9	8	10	9	4	4	7
u_3	5	1	8	6	9	10	3	5	7	10
Group	20	21	21	25	26	28	22	15	14	23

2.2 Multiplicative Utilitarian Strategy

For each item, the ratings given by users are multiplied and a ranked list of items is produced.

In [7] this strategy is adopted, in order to produce music recommendations.

	A	B	C	D	E	F	G	H	I	J
u_1	8	10	7	10	9	8	10	6	3	6
u_2	7	10	6	9	8	10	9	4	4	7
u_3	5	1	8	6	9	10	3	5	7	10
Group	280	100	336	540	648	800	270	120	84	420

2.3 Borda Count

Each item gets a number of points, according to the position in the list of each user. The least favorite item gets 0 points and a point is added each time the next item in the list is considered. If a user gave the same rating to more items, points are distributed. So, for example, items H and I were rated by user u_2 with the lowest rating and should "share" the lowest positions with 0 and 1 points, so both the items get $(0+1)/2=0.5$ points. A group preference is obtained by adding the individual points of an item.

This strategy was implemented in [3].

	A	B	C	D	E	F	G	H	I	J
u_1	4.5	8	3	8	6	4.5	8	1.5	0	1.5
u_2	3.5	7.5	2	6.5	5	7.5	6.5	0.5	0.5	3.5
u_3	2.5	0	5	3	6	7.5	1	2.5	4	7.5
Group	10.5	15.5	10	17	17	19.5	15.5	4.5	4.5	12.5

2.4 Copeland Rule

It is a form of majority voting that sorts items according to their *Copeland index*, which is calculated as the number of times in which an alternative beats the others, minus the number of times it loses against the other alternatives. In the example, item B beats item A, since both u_1 and u_2 gave it a higher rating.

The approach proposed in [8] proved that a form of majority voting is the most successful in a *requirements negotiation* context, in which the system resolves the existing conflicts between requirements and decides the ones that are implemented.

	A	B	C	D	E	F	G	H	I	J
A	0	+	-	+	+	+	+	-	-	0
B	-	0	-	0	-	0	0	-	-	-
C	+	+	0	+	+	+	+	-	-	+
D	-	0	-	0	-	+	-	-	-	-
E	-	+	-	+	0	+	+	-	-	-
F	-	0	-	-	-	0	-	-	-	-
G	-	0	-	+	-	+	0	-	-	-
H	+	+	+	+	+	+	+	0	0	+
I	+	+	+	+	+	+	+	0	0	+
J	0	+	+	+	+	+	+	-	-	0
Index	-2	+6	-3	+6	+1	+8	+4	-8	-8	-2

2.5 Plurality Voting Strategy

Each user votes for her/his favorite option. The alternative that receives the highest number of votes wins. If more than an alternative needs to be selected, the options that received the highest number of votes are selected.

This strategy was implemented and tested by [23, 22] in a TV domain.

	1	2	3	4	5	6
u_1	B, D, G	D, G	E	A	C	H
u_2	B, F	D, G	E	A, J	C	H, I
u_3	F, J	J	J	J	C	I
Group	B, F	D, G	E	A, J	C	H, I

2.6 Approval Voting

Each user can vote for as many items as she/he wants and a point is assigned to all the items a user likes. To show how the strategy works, we are going to suppose that each user votes for all the items with a rating above a certain threshold (let's say 5). A group preference is obtained by adding the individual points of an item.

When choosing the pages to recommend to a group, *Let's Browse* [13] evaluates if the page currently considered by the system matches with the user profile above a certain threshold and recommends the one that gets the highest score.

	A	B	C	D	E	F	G	H	I	J
u_1	1	1	1	1	1	1	1	1		1
u_2	1	1	1	1	1	1	1			1
u_3			1	1	1	1			1	1
Group	2	2	3	3	3	3	2	1	1	3

2.7 Least Misery Strategy

The rating assigned to an item for a group is the lowest rating expressed for that item by any of the members of the group. This strategy is usually used to model small groups, to make sure that every member is satisfied. A drawback of this strategy is that if the majority of the group really likes something, but one person does not, the item will not to be recommended to the group. This is what happens in the example for items B and G.

This strategy is used by *PolyLens* [17], in order to produce movie recommendations that satisfy the small groups handled by the system.

	A	B	C	D	E	F	G	H	I	J
u_1	8	10	7	10	9	8	10	6	3	6
u_2	7	10	6	9	8	10	9	4	4	7
u_3	5	1	8	6	9	10	3	5	7	10
Group	5	1	6	6	8	8	3	4	3	6

2.8 Most Pleasure Strategy

The rating assigned to an item for a group is the highest rating expressed for that item by a member of the group.

This strategy is adopted by [20] in a case-based group recommender system, proposed as a solution to the cold start problem.

	A	B	C	D	E	F	G	H	I	J
u_1	8	10	7	10	9	8	10	6	3	6
u_2	7	10	6	9	8	10	9	4	4	7
u_3	5	1	8	6	9	10	3	5	7	10
Group	8	10	8	10	9	10	10	6	7	10

2.9 Average Without Misery Strategy

The rating assigned to an item for a group is the average of the ratings assigned by each user for that item. All the items that were evaluated by a user with a rating under a certain threshold are not considered in the group model (in the example the threshold rating is 4).

In order to model the preferences of a group for each genre of music that can be played play in a gym, *MusicFX* [16] sums the individual ratings expressed by each user, discarding the ones under a minimum degree of satisfaction. As previously mentioned, in [18], the authors illustrate that an average is the best way to model individual preferences in a group model in different contexts and employ this strategy in their study.

	A	B	C	D	E	F	G	H	I	J
u_1	8	10	7	10	9	8	10	6	3	6
u_2	7	10	6	9	8	10	9	4	4	7
u_3	5	1	8	6	9	10	3	5	7	10
Group	20	-	21	25	26	28	-	15	-	23

2.10 Fairness Strategy

This strategy is based on the idea that users might be recommended something they do not like, as long as in the list of recommended items there is also something they like. This is done by allowing each user to choose her/his favorite item. If two items have the same rating, the choice is based on the other users' preferences. This is done until everyone has made a choice. Next, everyone chooses a second favorite item, usually starting from the person who chose last the first time.

If in the example we suppose that user u_1 chose first, she/he would select items B, D and G and would choose D, because it has the highest average considering the other users' ratings. Next u_2 would choose between items B and F and would choose F for the same reason. Then u_3 would choose item J (the item that was evaluated with the highest rating). Since everyone chose an item, it would be u_3's turn again and item E would be chosen. User u_2 would then choose item B, who has the highest rating along with F (which was already chosen). Then u_1 would choose item A, which is the one with the highest rating that was not chosen yet. Following this strategy, the sequence of item that models the group would be: D, F, J, E, B, A, C, H, I.

This strategy was adopted in the music recommender system proposed in [7].

	A	B	C	D	E	F	G	H	I	J
u_1	8	10	7	10	9	8	10	6	3	6
u_2	7	10	6	9	8	10	9	4	4	7
u_3	5	1	8	6	9	10	3	5	7	10

2.11 Most Respected Person Strategy (Dictatorship)

This strategy selects the items according to the preferences of the most respected person, using the preferences of the other users just in case more than an item received the same evaluation. The idea behind this strategy is that there are scenarios is which a group is guided/dominated by a person. In the following example, it is supposed that u_1 is the most respected person.

This strategy is used by *INTRIGUE* [2] that, in order to build a group model, advantages the preferences of a subset of users with particular needs. The studies conducted on the *G.A.I.N.* [19] system show that when people interact, a user or a small portion of the group influence the choices of the whole group. In [11], Jung develops an approach to identify long tail users, i.e., users who can be considered as expert group on a certain attribute. So, the ratings given by the long tail user groups are considered in order to provide a relevant recommendation to the non-expert user group, which are called short head group.

	A	B	C	D	E	F	G	H	I	J
u_1	8	10	7	10	9	8	10	6	3	6
u_2	7	10	6	9	8	10	9	4	4	7
u_3	5	1	8	6	9	10	3	5	7	10
Group	8	10	7	10	9	8	10	6	3	6

3. GROUP RECOMMENDATION WITH DETECTION OF GROUPS

Recently, we presented [6] a group recommender system, named *Predict&Cluster*, which automatically detects groups by clustering users. The system used *Additive Utilitarian* to model the individual preferences.

This section describes the different group recommender systems implemented for this study, which will vary from *Predict&Cluster* just on the strategy used by the group modeling task. By implementing the other tasks in the same way in all the systems, we will evaluate the impact of the different strategies on the performance of a system. The tasks performed by the systems are the following:

1. *Predictions of the missing ratings for individual users.* Predictions are built for each user with a User-Based Collaborative Filtering Approach.

2. *Detection of the groups.* Considering both the individual preferences expressed by each user and the predicted ratings, groups of similar users are detected with the k-means clustering algorithm.

3. *Group modeling.* Each system implements one of the group modeling strategies that could be applied to our context.

All the tasks are now be described in detail.

3.1 Predictions of the missing ratings for individual users.

The missing ratings are predicted for each user with a classic User-Based Nearest Neighbor Collaborative Filtering algorithm, presented in [21]. The algorithm predicts a rating p_{ui} for each item i that was not evaluated by a user u, by 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:

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

Values $\overline{r}_u$ and $\overline{r}_n$ indicate the mean of the ratings given by user u and user n. Similarity $userSim()$ among two users is calculated using the Pearson's correlation, which compares the ratings of the items rated by both the target user and the neighbor. Pearson's correlation among a user u and a neighbor n is given in Equation (2) (I_{un} is the set of items rated by both u and n).

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

The metric ranges between 1.0 and -1.0. Negative values do not increase the prediction accuracy [9], so they are discarded by the task.

3.2 Detection of the groups.

In order to respect the constraint imposed by the context, the set of users has to be partitioned into a number of groups equal to the number of recommendations that can be produced. Since in our application scenario groups do not exist, unsupervised classification (*clustering*) is necessary.

In [1], authors highlight that the k-means clustering algorithm is by far the most used clustering algorithm in recommender systems. In [6] we showed that by clustering users with k-means and by considering both the preferences expressed by a user and the ratings predicted by the previous task, the accuracy of a system can be improved. Therefore, this approach is also used in this study.

The output produced by the task is a partitioning of the users into groups (clusters), such that users with similar models (i.e., similar ratings for the same items) are in the same group and can receive the same recommendations.

3.3 Group modeling.

In order to create a model that represents the preferences of a group, we considered the strategies previously described in Section 2 and implemented all the ones that could be used in our context. In fact, there are some strategies that do not produce an explicit rating, but just a ranked list of the items evaluated by the group (i.e., the *Plurality Voting*, *Copeland Rule* and *Fairness* strategies), and there is a strategy (*Most Respected Person*) where just the ratings of the user who guides the group are considered (the idea of a "most respected person" is not meaningful in a context where a group of people is automatically detected). Even if the *Multiplicative Utilitarian* strategy was implemented, it could not be tested in our context. This happens because of the limit on the maximum number that can be calculated by a computer[1].

The other strategies, i.e., *Additive Utilitarian*, *Borda Count*, *Approval Voting*, *Least Misery*, *Most Pleasure* and *Average Without Misery*, were implemented.

As it can be noticed in the examples in Section 2, the group ratings produced by each strategy are in completely different scales of representation (i.e., in the examples individual ratings are expressed with a rating between 1 and 10, while group ratings can be much higher than 10). In order to evaluate how each group rating reflects the individual preferences, it is necessary to have both individual and group ratings are in the same domain. This can be obtained with a reduction:

$$group\_rating : max\_group\_rating = new\_group\_rating : max\_user\_rating$$

where:

- *group_rating* is the rating produced by a modeling strategy;
- *max_group_rating* is the maximum value of *group_rating* that can be obtained for an item;
- *max_user_rating* is the maximum rating that a user can express for an item.

So a *new_group_rating* can be obtained calculating:

$$new\_group\_rating = \frac{group\_rating \cdot max\_user\_rating}{max\_group\_rating}$$

The reduction has been adapted for some strategies. In fact, considering *Approval Voting*, *max_group_rating* is equal to the number of users in a group, since an item might get a point from every user. In the *Borda Count* strategy, *max_group_rating* cannot take the same value for all the items, since the strategy ranks each user's preferences. Therefore, a mapping to the range of values of the individual ratings is necessary. The values considered in the mapping are:

group_rating, i.e., the rating produced by the modeling strategy;

min_group_rating, i.e., the lowest value of *group_rating* that can be obtained;

max_group_rating, i.e., the highest value of *group_rating* that can be obtained;

min_user_rating, i.e., the lowest rating that a user can express;

[1] A 64 bits machine cannot calculate numbers higher than 2^{52}. That would mean that even if the strategy was tested with a very small dataset of 55 users and they all gave a very small rating for an item, like 2, an overflow would occur ($2^{55} > 2^{52}$).

max_user_rating, i.e., the highest rating that a user can express.

In order to produce the mapping, we first calculate how wide each range is:

$$modeled\_ratings\_span = max\_group\_rating - min\_group\_rating$$

$$user\_ratings\_span = max\_user\_rating - min\_user\_rating$$

Then, we convert the range produced by the strategy into a 0-1 range:

$$modeled\_value = \frac{group\_rating - min\_group\_rating}{modeled\_ratings\_span}$$

The 0-1 range is then converted into the desired range:

$$new\_group\_rating = min\_user\_rating + (modeled\_value \cdot user\_ratings\_span)$$

4. EXPERIMENTAL FRAMEWORK

This section presents the framework built for the experiments.

4.1 Experimental Setup

To conduct the experiments, we adopted the MovieLens-1M dataset.

The number of neighbors used by the first task to predict the ratings is 100 (see [5] for the details of the experiments that allowed to set the value).

The clusterings with k-means were created using a testbed program called KMlocal [12], which contains a variant of the k-means algorithm, called *EZ Hybrid*, that was chosen because it returned a lowest average distortion.

The RMSE values obtained by each system have been compared, by considering different numbers of groups to detect (20, 50, 200 and 500 groups). Moreover, we evaluated the performance obtained by considering a single group with all the users and by calculating the predictions for each user.

Note that if a strategy presents a threshold value (i.e., *Approval Voting* and *Average Without Misery*), all the possible values are tested.

RMSE was chosen to compare the systems because it is widely used, allows to evaluate results through a single number and emphasizes large errors.

In order to evaluate if two RMSE values returned by two experiments are significantly different, independent-samples two-tailed Student's t-tests have been conducted. In order to make the tests, a 5-fold cross-validation was preformed.

4.2 Dataset and Data Preprocessing

The MovieLens-1M[2] dataset contains 1 million ratings, given by 6040 users for 3900 movies. Our framework uses only the file `ratings.dat`, which contains the user ratings. The file was preprocessed by mapping the feature *UserID* into a new set of IDs between 0 and 6039 to facilitate the computation with data structures. In order to conduct the cross-validation, the dataset was split into five subsets with a random sampling technique (each subset contains 20% of the ratings).

[2]http://www.grouplens.org/

Table 1: RMSE values obtained by using the different group modeling strategies.

	1 group	20 groups	50 groups	200 groups	500 groups	6040 groups
AU	0.9895	0.9554	0.9435	0.9395	0.9385	0.9102
AV [thr=1]	1.7854	1.7634	1.7558	1.7573	1.7629	0.9102
AV [thr=2]	1.6841	1.6112	1.6025	1.6057	1.6193	0.9102
BC	1.0767	1.0667	1.0624	1.0596	1.0570	0.9102
LM	2.9878	2.4782	2.1972	1.8868	1.7024	0.9102
MP	1.7977	1.6786	1.5796	1.4648	1.3735	0.9102

4.3 Metrics

The quality of the predicted ratings was measured through the Root Mean Squared Error (RMSE). The metric compares each rating r_{ui}, expressed by a user u for an item i in the test set, with the rating p_{gi}, predicted for the item i for the group g 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. In order to compare if two RMSE values returned by two experiments are significantly different, independent-samples two-tailed Student's t-tests have been conducted. These tests allow us to reject the null hypothesis that two values are statistically the same. So, a two-tailed test will evaluate if an RMSE value is significantly greater or significantly smaller than another RMSE value. Since each experiment was conducted five times, the means M_i and M_j of the RMSE values obtained by two systems i and j are used to compare the systems and calculate a value t:

$$t = \frac{M_i - M_j}{s_{M_i - M_j}}$$

where

$$s_{M_i - M_j} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$$

s^2 is the variance of the two samples, n_1 and n_2 indicate the number of values considered to build M_1 and M_2 (in our case both are equal to 5, since experiments were repeated five times). In order to determine the $t - value$ that indicates the result of the test, the degrees of freedom have to be determined:

$$\text{d.f.} = \frac{(s_1^2/n_1 + s_2^2/n_2)^2}{(s_1^2/n_1)^2/(n_1 - 1) + (s_2^2/n_2)^2/(n_2 - 1)}$$

Given t and $d.f.$, the $t - value$ (i.e., the results of the test), can be obtained in a standard table of significance as: $t(d.f.) = t - value$. The $t - value$ derives the probability p that there is no difference between the two means. Along with the result of a t-test, the standard deviation SD of the mean is presented.

4.4 Experimental results

Figure 1 and Table 1 show the RMSE values obtained by each system.

Note that the results obtained for the *Average Without Misery* strategy with all the thresholds and the *Approval Voting* strategy with thresholds 3 and 4 have been omitted, since a small portion of the test set could be considered (less

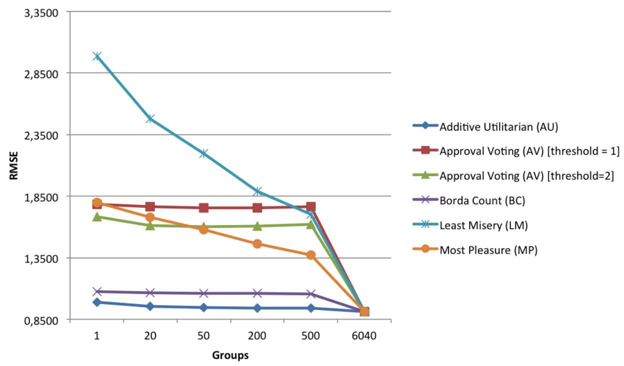

Figure 1: RMSE values obtained by using the different group modeling strategies.

than 10%) and results were not reliable. In fact, the *Average Without Misery* strategy discards a group rating if at least a user has evaluated an item with a rating lower than the threshold. Even with a small value, like 2, the vast majority of the items is not modeled by the strategy, since groups are large (in fact, the larger is the group, the higher is the probability that there is at least one user who did not like the item). Strategy *Approval Voting* with thresholds 3 and 4 could not be evaluated, because by considering only items with a high rating (i.e., above 3), too many ratings were removed from the model.

Results show that, in every clustering, *Additive Utilitarian* is the strategy that best models the groups. This allows to derive two interesting properties:

- since this scenario deals with a limited number of recommendations, the system works with large groups. Therefore, an average, which is a single value that typifies a set of values, is best way to combine the ratings in this context;
- a model created with an average of the individual preferences represents the centroid of the cluster, i.e., a super-user that connects the users of a group.

This means that the clustering and the group modeling tasks are linked. In fact, *Additive Utilitarian* is able to exploit two properties of the groups handled in this context, i.e., its size and the fact that all the users in a group are connected to a centroid.

It can be noticed that the strategies that advantage a user (i.e., *Least Misery* and *Most Pleasure*) do not produce accurate models. This is also true for *Approval Voting*, which does not consider the ratings under a certain threshold. By using these strategies, accuracy is affected, since a user is not considered or a part of her/his preferences is not included in the model.

Student's t-test allowed to validate the results previously presented. All the tests returned a significant difference among the results. For readability reasons, we present just the test that involves the strategy with the results closer to *Additive Utilitarian* (AU), i.e., *Borda Count* (BC).

With 1 group, there is a significant difference between AU ($M = 0.9895$, $SD = 0.00$) and BC ($M = 1.0767$, $SD = 0.00$); $t(6.91) = 230.75$, $p < 0.05$.

With 20 groups, there is a significant difference between AU ($M = 0.9554$, $SD = 0.00$) and BC ($M = 1.0667$, $SD = 0.00$); $t(7.03) = 55.53$, $p < 0.05$.

With 50 groups, there is a significant difference between AU ($M = 0.9435$, $SD = 0.00$) and BC ($M = 1.0624$, $SD = 0.00$); $t(5.04) = 139.27$, $p < 0.05$.

With 200 groups, there is a significant difference between AU ($M = 0.9395$, $SD = 0.00$) and BC ($M = 1.0596$, $SD = 0.00$); $t(7.54) = 192.73$, $p < 0.05$.

With 500 groups, there is a significant difference between AU ($M = 0.9385$, $SD = 0.00$) and BC ($M = 1.0570$, $SD = 0.00$); $t(7.98) = 225.65$, $p < 0.05$.

5. CONCLUSIONS AND FUTURE WORK

This paper compared group modeling strategies in a group recommender system that detects groups. Throughout this work, several lessons have been learned:

- the strategy used to model groups strongly affects the accuracy of the group recommendations that are produced and there is a statistical difference among the results produced by each strategy;
- in this context, *Additive Utilitarian* is the strategy that best models the groups. In fact, by averaging individual preferences, a group model has the same role that the centroid has when users are clustered (i.e., individual preferences are closer to the average than to any other value).
- if the number of groups is small with respect to the number of users, averaging individual preferences produces group preferences that equally consider each user. If a strategy that advantages a part of the group is embraced in this context, the accuracy of the group recommendations is affected;
- only a subset of strategies can be used in this context and they work only if not too many preferences are discarded from the group modeling task.

Future work will explore the structure of the groups furthermore, by analyzing which properties of a group affect the accuracy of a system.

6. REFERENCES

[1] X. Amatriain, A. Jaimes, N. Oliver, and J. M. Pujol. Data mining methods for recommender systems. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 39–71. Springer, Boston, MA, 2011.

[2] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso. Intrigue: Personalized recommendation of tourist attractions for desktop and hand held devices. *Applied Artificial Intelligence*, 17(8-9):687–714, 2003.

[3] L. Baltrunas, T. Makcinskas, and F. Ricci. Group recommendations with rank aggregation and collaborative filtering. In X. Amatriain, M. Torrens, P. Resnick, and M. Zanker, editors, *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*, pages 119–126. ACM, 2010.

[4] L. Boratto and S. Carta. State-of-the-art in group recommendation and new approaches for automatic identification of groups. In A. Soro, E. Vargiu, G. Armano, and G. Paddeu, editors, *Information Retrieval and Mining in Distributed Environments*, volume 324 of *Studies in Computational Intelligence*, pages 1–20. Springer Berlin Heidelberg, 2011.

[5] L. Boratto and S. Carta. Exploring the ratings prediction task in a group recommender system that automatically detects groups. In *IMMM 2013, The Third International Conference on Advances in Information Mining and Management*, pages 36–43, 2013.

[6] L. Boratto and S. Carta. Using collaborative filtering to overcome the curse of dimensionality when clustering users in a group recommender system. In *Proceedings of 16th International Conference on Enterprise Information Systems (ICEIS)*, 2014.

[7] I. A. Christensen and S. N. Schiaffino. Entertainment recommender systems for group of users. *Expert Systems with Applications*, 38(11):14127–14135, 2011.

[8] A. Felfernig, C. Zehentner, G. Ninaus, H. Grabner, W. Maalej, D. Pagano, L. Weninger, and F. Reinfrank. Group decision support for requirements negotiation. In L. Ardissono and T. Kuflik, editors, *Advances in User Modeling - UMAP 2011 Workshops, Girona, Spain, July 11-15, 2011, Revised Selected Papers*, volume 7138 of *Lecture Notes in Computer Science*, pages 105–116. Springer, 2012.

[9] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 230–237, New York, NY, USA, 1999. ACM.

[10] 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*, volume 4321 of *Lecture Notes in Computer Science*, pages 596–627. Springer, 2007.

[11] J. J. Jung. Attribute selection-based recommendation framework for short-head user group: An empirical study by movielens and imdb. *Expert Systems with Applications*, 39(4):4049–4054, Mar. 2012.

[12] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):881–892, 2002.

[13] H. Lieberman, N. W. V. Dyke, and A. S. Vivacqua. Let's browse: A collaborative web browsing agent. In *IUI*, pages 65–68, 1999.

[14] J. Masthoff. Group recommender systems: Combining individual models. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 677–702. Springer, 2011.

[15] J. McCarthy. Pocket RestaurantFinder: A situated recommender system for groups. In *Workshop on Mobile Ad-Hoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems*, 2002.

[16] J. F. McCarthy and T. D. Anagnost. Musicfx: An arbiter of group preferences for computer supported collaborative workouts. In S. E. Poltrock and J. Grudin, editors, *CSCW '98, Proceedings of the ACM 1998 Conference on Computer Supported Cooperative Work, Seattle, WA, USA, November 14-18, 1998*, pages 363–372. ACM, 1998.

[17] M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl. Polylens: A recommender system for groups of users. In W. Prinz, M. Jarke, Y. Rogers, K. Schmidt, and V. Wulf, editors, *Proceedings of the Seventh European Conference on Computer Supported Cooperative Work, 16-20 September 2001, Bonn, Germany*, pages 199–218. Kluwer, 2001.

[18] T. Pessemier, S. Dooms, and L. Martens. Comparison of group recommendation algorithms. *Multimedia Tools and Applications*, pages 1–45, 2013.

[19] S. Pizzutilo, B. D. Carolis, G. Cozzolongo, and F. Ambruoso. Group modeling in a public space: Methods, techniques and experiences. In *Proceedings of WSEAS AIC 05*, Malta, 2005. ACM.

[20] L. Quijano-Sánchez, D. G. Bridge, B. Díaz-Agudo, and J. A. Recio-García. A case-based solution to the cold-start problem in group recommenders. In B. Díaz-Agudo and I. Watson, editors, *Case-Based Reasoning Research and Development - 20th International Conference, ICCBR 2012, Lyon, France, September 3-6, 2012. Proceedings*, volume 7466 of *Lecture Notes in Computer Science*, pages 342–356. Springer, 2012.

[21] J. B. Schafer, D. Frankowski, J. L. Herlocker, and S. Sen. Collaborative filtering recommender systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web, Methods and Strategies of Web Personalization*, volume 4321 of *Lecture Notes in Computer Science*, pages 291–324. Springer, 2007.

[22] C. Senot, D. Kostadinov, M. Bouzid, J. Picault, and A. Aghasaryan. Evaluation of group profiling strategies. In T. Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 2728–2733. IJCAI/AAAI, 2011.

[23] C. Senot, D. Kostadinov, M. Bouzid, J. Picault, A. Aghasaryan, and C. Bernier. Analysis of strategies for building group profiles. In P. D. Bra, A. Kobsa, and D. N. Chin, editors, *User Modeling, Adaptation, and Personalization, 18th International Conference, UMAP 2010, Big Island, HI, USA, June 20-24, 2010. Proceedings*, volume 6075 of *Lecture Notes in Computer Science*, pages 40–51. Springer, 2010.