

Group Recommendations with Rank Aggregation and Collaborative Filtering

Linas Baltrunas
Free University of
Bozen-Bolzano,
Piazza Domenicani 3,
Bolzano, Italy
lbaltrunas@unibz.it

Tadas Makcinskas
Free University of
Bozen-Bolzano,
Piazza Domenicani 3,
Bolzano, Italy
tmakcinskas@unibz.it

Francesco Ricci
Free University of
Bozen-Bolzano,
Piazza Domenicani 3,
Bolzano, Italy
fricci@unibz.it

ABSTRACT

The majority of recommender systems are designed to make recommendations for individual users. However, in some circumstances the items to be selected are not intended for personal usage but for a group; e.g., a DVD could be watched by a group of friends. In order to generate effective recommendations for a group the system must satisfy, as much as possible, the individual preferences of the group's members.

This paper analyzes the effectiveness of group recommendations obtained aggregating the individual lists of recommendations produced by a collaborative filtering system. We compare the effectiveness of individual and group recommendation lists using normalized discounted cumulative gain. It is observed that the effectiveness of a group recommendation does not necessarily decrease when the group size grows. Moreover, when individual recommendations are not effective a user could obtain better suggestions looking at the group recommendations. Finally, it is shown that the more alike the users in the group are, the more effective the group recommendations are.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Information Filtering

General Terms

Algorithms, Experimentation

Keywords

Group recommender system, rank aggregation, collaborative filtering

1. INTRODUCTION

Recommender Systems (RSs) are software tools and techniques suggesting to a user a well selected set of items match-

ing the user's taste and preferences [18, 1, 4]. The suggestions relate to various decision-making processes, such as what items to buy, what music to listen to, or what on-line news to read. They are widely used in Web-based e-commerce applications to help online users to choose the most suitable products, e.g. movies, CDs, books, or travels.

The large majority of RSs are designed to make recommendations for individual users. Since recommendations are usually personalized, different users receive diverse suggestions. However, in some circumstances the items to be selected are not intended for personal usage but for a group of users; e.g., a DVD could be watched by a group of friends or in a family. These groups can vary from stable groups to ad hoc groups requiring recommendations only occasionally.

For this reason some recent works have addressed the problem of identifying recommendations "good for" a group of users, i.e., trying to satisfy, as much as possible, the individual preferences of all the group's members [8]. Group recommendation approaches are either based on the generation of an integrated group profile or on the integration of recommendations built for each member separately (see section 2). Group RSs have been designed to work in different domains: web/news pages [17], tourism [15, 13], music [6, 14], TV programs and movies [16, 21].

A major issue in this research area relates to the difficulty of evaluating the effectiveness of group recommendations, i.e., comparing the generated recommendations for a group with the true preferences of the individual members. One general approach for such an evaluation consists of interviewing real users. In this approach there are two options: either to acquire the users' individual evaluations for the group recommendations and then integrate (e.g., averaging) these evaluations into a score that the group "jointly" assigns to the recommendations; or to acquire directly a joint evaluation of the group for the recommendations. In the first case one must decide how the individual evaluations are integrated; this is problematic as different methods will produce different results and there is no single best way to perform such an integration. Another difficulty, which is common to both options, as was observed by [12], is related to the fact that the satisfaction of an individual is likely to depend on that of other individuals in the group (emotional contagion). Moreover, on-line evaluations can be performed on a very limited set of test cases and cannot be used to extensively test alternative algorithms.

A second approach consists of performing off-line evaluations, where groups are sampled from the users of a tra-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

RecSys2010, September 26–30, 2010, Barcelona, Spain.

Copyright 2010 ACM 978-1-60558-906-0/10/09 ...\$10.00.

ditional (i.e., single-user) RS. Group recommendations are offered to group members and are evaluated independently by them, as in the classical single user case, by comparing the predicted ratings (rankings) with the ratings (rankings) observed in the test set of the user. The group recommendations are generated to suit simultaneously the preferences of all the users in the group and our intuition suggests that they cannot be as good as the individually tailored recommendations. We observe that using this evaluation approach, in order to test the effectiveness of a group recommendation, we do not need the joint group evaluations for the recommended items, and we can reuse the most popular data sets (e.g, MovieLens or Netflix) that contain just evaluations (ratings) of individual users.

In this paper we follow the second approach evaluating a set of novel group recommender techniques based on rank aggregation. We first generate synthetic groups (with various criteria), then we build recommendation lists for these groups, and finally we evaluate these group recommendations on the test sets of the users. The group recommendations are built in two steps: first a collaborative filtering algorithm produces individual users' rating predictions for the test items and consequently individual ranking predictions based on these rating predictions are generated. Then a rank aggregation method generates a joint ranking of the items recommended to the group by integrating the individual ranking predictions computed in the previous step. We then measure, for each group member, how good this integrated ranking is, and if this is worse than the initial individual ranking built by the system by taking into account only the ratings contained in the profile of the user. We performed an analysis of the generated group recommendations (ranking) varying the size of the groups, the inner group members similarity, and the rank aggregation mechanism.

As we mentioned above, intuition suggests that aggregating individual rankings, i.e., mediating the potentially contrasting preferences of the group members, should result in a decrease of the recommendation effectiveness. Moreover, one can also conjecture that the larger is the number of people in the group the harder it is to find a consensus among them.

In this work, we show that these intuitions are not always correct. In fact, we show that in certain cases group recommendations with rank aggregation can be more effective than individual recommendations, when effectiveness is measured using normalized discounted cumulative gain (nDCG), a popular IR metric measuring the quality of the ranking produced by a system. We observe that this happens when the individual recommendations are not particularly good, i.e., when the recommender system is not able to make good personalized recommendations. We show that this happens often in real scenarios. Moreover, we confirm the intuition that the effectiveness of the group recommendations raises when the group members are more similar.

It is worth noting that in this paper we focus on the goodness of the predicted *ranking* of the recommendations rather than in evaluating the accuracy of the predicted ratings' values. We believe that especially for group recommendations it is more important to understand if the RS correctly sorts the proposed items, hence suggests to the group first the items more suitable for all the group members. This approach follows a new trend of evaluating recommender systems [19]. Moreover, we observe that the rank aggregation

problem has been largely studied and the notion of optimal rank, i.e., the Kemeny optimal one, has been defined [7]. Some of the rank aggregation techniques that we have used (e.g., Spearman footrule optimal aggregation) are proved to be close to the Kemeny optimal one and therefore our results show important and fundamental properties of the best solutions to group recommendation via rank aggregation.

This paper is structured as follows. Section 2 overviews related researches. Section 3 defines our approach based on rank aggregation. Section 5 provides the experimental evaluation of the approach and finally Section 6 draws the conclusions and describes some future work.

2. RELATED WORK

The major part of the research on group recommendation investigated the core algorithms used for generating the group recommendations. Different strategies are available and two main approaches have been proposed [8]. The first consists of creating a joint user profile for all the users in the group and then performing a recommendation for this artificial user represented by the group profile [14, 21]. This would provide a recommendation for the group that is based on all the user profiles and in some way represents the group interests mediated in the group profile. The second approach aggregates the recommendations for each individual member into a single group recommendation list. In this case, first the recommendations for each individual group member are created independently, and then these ranked lists of items, one for each individual user, are aggregated into a joint group recommendation list. Berkovsky and Freyne [3] compared the two approaches in a recipe recommendation problem and found that the first one performs slightly better. In their study the second approach is implemented by computing an item rating prediction for a group as a domain-dependent linear combination of the predictions for the group members. Actually, this is only one of the many possible approaches for aggregating predictions and an extensive comparison of the two approaches is still missing in the literature.

In our work we are concerned with recommendations generated using this second approach, i.e., aggregating individual recommendations. The core issue in this case is how to aggregate into a single group recommendations' list the ranked lists of recommendations produced for each group member. And then the second problem is related to the estimation of how this integrated ranking would be evaluated by the group members.

Masthoff [11] addressed the related problem of how subjects aggregate recommendation lists, i.e., how a person would select recommendations for a group balancing the preferences of the group members (three, in their experiments), that are expressed by the ratings of these members for 10 options. The subjects were asked which items (clips) the three should view as a group, given that they only had time to see 1, 2, 3, ... or 7. The goal was primarily to understand how humans aggregate group preferences and hence to select the aggregation method that is best evaluated by the users. That study concluded that there is evidence that human subjects use in particular Average, Average Without Misery, and Least Misery (see [11] for the definition of the aggregation strategies). Average and Least Misery are two strategies that we have tested in our experiments. We note however, that [11] did not evaluate how group members

evaluate group recommendations but how subjects integrate recommendations for a group.

Another related work in the area of web search [7] deals with rank aggregation methods for the web, and presents a theoretical groundwork for building and evaluating rank aggregation methods. Kemeny optimal aggregation is introduced, and since this is computationally intractable other aggregation methods are defined, showing that their produced rankings are close to the optimal one. We have adopted that approach and used rank aggregation methods for combining the ranking independently produced for each group member by a collaborative filtering RS. The major technical difficulties that we faced are related to the specific characteristics of our application. In fact, in their case the best integrated ranking is defined as the Kemeny optimal one, i.e., that minimizing the (average) Kendall tau distance from the *predicted* individual recommendation lists.

In the recommendation scenario the ultimate goal is not simply to correctly aggregate the individual recommendation lists. In fact, that aggregation may be optimal, i.e., as close as possible to the individual recommendation lists, but far from the true users' preferences, i.e., those observed in the test data. For instance, suppose that there are three items p_1, p_2 and p_3 , and a group of users all rank these items as $[p_1, p_2, p_3]$. Imagine that the RS makes a systematic error and predicts the ratings for the users in the group in such a way that the generated ranking is $[p_3, p_2, p_1]$, i.e., it is the reverse order of the correct one. Then the predicted ranking for the group is also equal to $[p_3, p_2, p_1]$. Clearly, this is the Kemeny optimal aggregated ranking (since all of them are equal) but it is a wrong group recommendation (because the predictions were wrong) when it is compared to the true preferred ranking of the group. To avoid this problem, in our experimental evaluation the aggregated group ranking is compared to the individuals' optimal rankings that we derive from the ratings in the test set.

3. RANK AGGREGATION FOR GROUP RECOMMENDATIONS

Our group recommendation method is based on the ordinal ranking of items, i.e., the result of a group recommendation process is an ordered list of items. To generate the ranking we use rank aggregation methods, taking a set of predicted ranked lists, one for each group member, and producing one combined and ordered recommendations' list.

Most of the available rank aggregation methods are inspired by results obtained in Social Choice Theory [2]. This theory studies how individual preferences could be aggregated to reach a collective consensus. Social choice theorists are concerned with combining ordinal rankings rather than ratings. Thus, they search for a societal preference result, where output is obtained combining several people ordered preferences on the same set of choices. For instance, in elections a number of ranked candidates, provided by voters, is collected and aggregated into one final ranked electors' list. Arrow [2] proved that this aggregation task is impossible, if the combined preferences must satisfy a few compelling properties. His theorem states that there cannot be one perfect aggregation method, and this allows for a variety of different aggregation methods to exist.

In this paper we will discuss a few different rank aggregation methods. There are two known ways to build a ranked

list of recommendations for a group by using aggregation methods. The first approach can be applied to lists of items where a prediction of the user rating for each item is available. This approach computes first the group score for an item, which is a kind of prediction of the joint group rating, by integrating the predicted ratings for the item in the various input lists. Then the group score for the item is used to rank the items for the whole group. The other approach uses only the ranked lists produced for each individual and then in order to get the final group ranking applies aggregation methods on these lists.

Creating recommendations using rank aggregation methods addresses the problem of finding "consensus" ranking between alternatives, given the individual ranking preferences of several judges [7]. The appropriate rank aggregation method usually depends on the ranking distance to optimize [7] and also on the properties of the lists that are aggregated. The aggregated list that minimizes the average Kendall tau distance from the input lists is called the Kemeny optimal aggregation. The Kendall tau distance counts the number of pair-wise disagreements between two lists. Kemeny optimal aggregation is the unique method that simultaneously satisfy natural and important properties of rank aggregation functions, called neutrality and consistency in the social choice literature (also known as Condorcet property) [20]. However, it was proved in [7] that such aggregation is NP-Hard and, therefore, we consider rank aggregation methods that approximate Kemeny optimal aggregation. These methods work with full lists of items, i.e., the aggregated lists contain the same items. We note that in our case this is not a limitation as we use collaborative filtering based on latent factor model [9]. Factor models can return ratings' predictions for all the unrated items and therefore can generate a full list of ranked items for all the users.

We now describe the rank aggregation methods that we use. All these methods take as input a set of items' permutations $g = \{\sigma_1, \dots, \sigma_{|g|}\}$, one for each group member, and produce a new permutation σ_g , i.e., the recommended ranking for the group g . σ_u is the permutation of the items $I = \{1, \dots, n\}$, representing the ranked list recommendations for user u ; $\sigma_u(j)$ is the position of item j in this list. For instance, $\sigma_u(j) = 1$ means that the item j is in the top position in the ranked list of recommendations for user u .

Spearman footrule aggregation finds an aggregation that minimizes the average Spearman footrule distance to the input rankings. The Spearman footrule distance between two lists is the sum, over all the items i of the absolute difference between the rank positions of i in the two lists. For instance, if σ_u, σ_v are two permutations of the items in I their Spearman footrule distance is: $F(\sigma_u, \sigma_v) = \sum_{i \in I} |\sigma_u(i) - \sigma_v(i)|$.

This aggregation method produces a ranking having average Kendall tau distance from the integrated rankings less than twice the average Kendall tau distance of the Kemeny optimal aggregation from the same rankings [7]. This aggregation is computed by finding a minimum cost perfect matching in a particular bipartite graph. Computations are made on the weighted complete bipartite graph (C, P, W) as follows. The first set of nodes C denotes the set of items to be ranked. The second set of nodes P denotes the number of available positions. The weight $W(c, p)$ is the total footrule distance (from all the individual user rankings) of the ranking that places element c at position p , and is given

by $W(i, p) = \sum_{u \in g} |\sigma_u(i) - p|$. We refer to [7] for additional details on this method.

In **Borda count** aggregation method each individual item in a group member’s ranked list is awarded with a score which is given according to its position in the list, $score_u(i) = n - \sigma_u(i) + 1$. The lower is the item position in the list (i.e., the larger is the value of $\sigma_u(i)$) the smaller is the score. Finally, the score points for the users in the group g , are added up, $score_g(i) = \sum_{u \in g} score_u(i)$, and are used to produce the aggregated ranking (in decreasing group score value). Recently, it was shown by Coppersmith et al. [5] that the Borda’s method produces aggregated lists that have average Kendall tau distance from the integrated rankings in g less than five times the average Kendall tau distance of the Kemeny optimal aggregation from the rankings in g .

The next two aggregation methods require that the items in the lists to aggregate are sorted according to (predicted) ratings. Since these lists represent recommendations for users the ratings are those predicted by the recommender system. The output of these methods is again a ranked list of items but the ranking is computed by using an aggregation method that uses the (predicted) ratings.

In **Average** aggregation the item i group score is equal to the average of the predicted ratings for the individuals, i.e., $score_g(i) = \frac{\sum_{u \in g} \hat{r}_{ui}}{|g|}$, where \hat{r}_{ui} is the predicted rating for user u , item i combination. Then the ranking is computed accordingly (decreasing values of group score).

Least Misery strategy also uses the predicted ratings of the items in the individual recommendation lists. In this method the group score for item i is equal to the smallest predicted rating for i in the group, i.e., $score_g(i) = \min_{u \in g} \{\hat{r}_{ui}\}$. Thus, each item is predicted to be liked by the group as the less satisfied member, and again in the integrated group ranking the items are ordered according to these scores.

As a baseline we also consider the **Random** aggregation method; it returns a random permutation of the items.

4. EXPERIMENTAL SETUP

In the experimental evaluation we measure the effectiveness of the proposed group recommendation techniques. For each member of a group, we compare the effectiveness of the ranked recommendations generated for his group with the effectiveness of those computed only for him. Effectiveness of a ranking is computed with nDCG, as it will be illustrated below. We deliberately avoided to compute the joint effectiveness of the group recommendations for the group because, as we discussed in the introduction, it requires to define an arbitrary aggregation formula integrating the effectiveness of the group recommendation for all the group members.

To conduct the experiments we first generated artificial groups of users (see Subsection 4.1 for the details). Then, for evaluating the goodness of the recommendations (both individual and group) we used the standard approach to divide the dataset into two parts: the training set and the testing set. Approximately 60% (randomly chosen) of the ratings from the user profile were assigned to the training set and the rest 40% to the testing set. We used collaborative filtering to generate individual predictions for each user (see Subsection 4.2 for the details). Then, these predictions were aggregated into the group recommendations using the

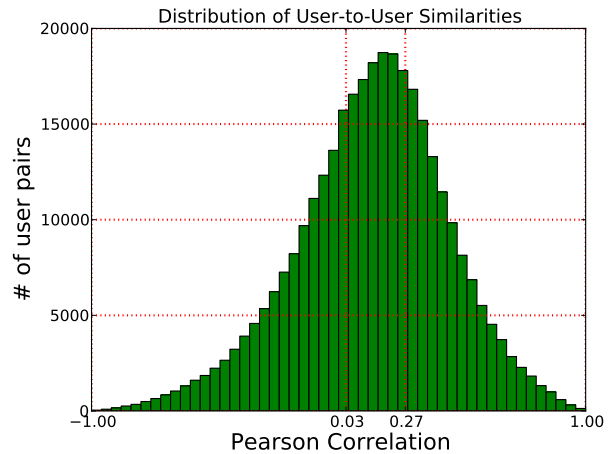


Figure 1: User-to-user similarity distribution.

previously described methods. Finally, in the last step we computed the recommendation lists (individual and group) ranking precision using Normalized Discounted Cumulative Gain (nDCG) measure (see Subsection 4.3).

4.1 Data Set and Group Generation

To conduct the experiments, user groups of varying sizes and similarities were generated by sampling MovieLens data set, which contains 100K ratings for 1682 movies by 943 users (the precise number of generated groups depends on the experiments and it is reported later on). We considered groups containing two, three, four and eight users. Moreover, we distinguished between *random groups* and groups with *high inner group similarity*. These are cases that are common in a real life situations. Groups with highly similar members represent people with common tastes, such as groups of friends. Whereas, random groups represent people without any social relations, such as random people taking the same bus.

Groups with high inner group similarity are defined as those containing users with user-to-user similarity higher than 0.27; where similarity is computed using Pearson correlation coefficient (PCC). In our data set 33% of all the possible user pairs have similarity higher than that threshold of 0.27 (see Figure 1). Random groups were formed without considering any restriction on the user-to-user similarity. That led to a lower overall inner group similarity. For example, the average similarity of the users in these random groups is 0.132, whereas the average similarity of the users in the groups with high inner group similarity is 0.456.

When forming the groups, in order to measure if two users are similar or not we decided to consider only pairs of users that have rated at least 5 common items. This is a common practice in memory based CF literature and assures that the computed similarity value is reliable and the correlation is not high or low just by chance, i.e., because there is a perfect agreement or disagreement on a small set of items.

4.2 Recommendation Lists for a Group

To compute rating predictions for each user and generate individual recommendations’ lists we used a popular model based collaborative filtering approach using matrix factor-

ization with gradient descent optimization [9]. Hence, in our experimental setup individual predictions are computed using Singular Value Decomposition latent factor model with 60 factors. Using this prediction method for each user we generated a ranked list of recommendations containing all the items that are not present in the user’s training set of any group member. Then, as we have discussed earlier, our group recommendation algorithm takes as input either these individual ranked lists of items’ recommendations or the predicted ratings for each user in the group, and returns a ranked list of recommendations for the whole group. The individual recommendations are aggregated using the five methods described in Section 3.

4.3 Effectiveness of a Ranked List of Recommendations

For evaluating the goodness of a ranked list of recommendations we use Normalized Discounted Cumulative Gain (nDCG), a standard IR measure [10]. Let p_1, \dots, p_l be a ranked list of items produced as an individual or group recommendation. Let u be a user and r_{up_i} the true rating of the user u for the item p_i (ranked in position i , i.e., $\sigma_u(p_i) = i$). Discounted Cumulative Gain (DCG) and normalized DCG (nDCG) at rank k are defined respectively as:

$$DCG_k^u = r_{up_1} + \sum_{i=2}^k \frac{r_{up_i}}{\log_2(i)} \quad (1)$$

$$nDCG_k^u = \frac{DCG_k^u}{IDCG_k^u} \quad (2)$$

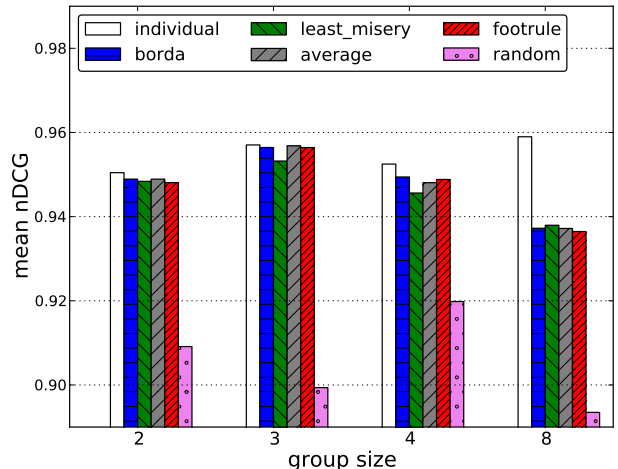
where IDCG is the maximum possible gain value for user u that is obtained with the optimal re-order of the k items in p_1, \dots, p_k .

To compute nDCG we need to know the true user rating for all the items in the recommendation list. Actually, when the test set (items rated by the users) contains only some of the items ranked in the recommendation list one must update the above definition. In our experiments we computed nDCG on all the items in the test set of the user sorted according to the ranking computed by the recommendation algorithm (individual or group recommendations). In other words, we compute nDCG on the projection of the recommendation list on the test set of the users. For example, imagine that $r = [1, 4, 5, 8, 3, 7, 6, 2, 9]$ is a ranked list of recommendations for a group. Since this is a group recommendation list, as we observed above, none of the items in this list occurs in the training set of any group member. Moreover, suppose that the user u test set consists of eight items $\{1, 4, 7, 8, 9, 12, 14, 20\}$. In such case, we would compute nDCG on the ranked list $[1, 4, 8, 7, 9]$.

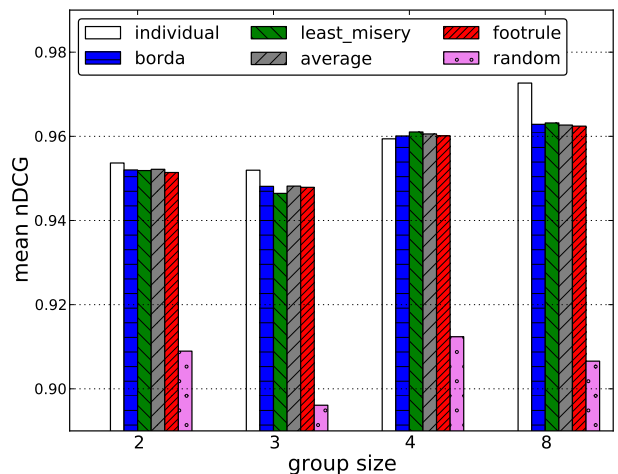
5. EXPERIMENTAL RESULTS

5.1 Effectiveness of Group Recommendations

In the first experiment we compare the effectiveness of the group and individual recommendations when varying the aggregation method and the group size. We conducted this experiment for random groups and groups with high inner group similarity. Our initial hypothesis was that the effectiveness of the group recommendation, in both cases, should decrease as the group size increases. In fact, our intuition says that, even without taking into account the



(a) random groups



(b) high similarity groups

Figure 2: Effectiveness of group recommendation with rank aggregation techniques.

group composition, it is harder to build good recommendations for larger groups as it gets harder to find a consensus among many, potentially different preferences. Moreover, when building recommendations for large groups we do not consider items that are already experienced (are in the training set) by any of the user in the group. This is likely to discard the most popular items, making it harder to make “everybody likes this” type of recommendations.

Figure 2 shows the results of our experiments using random groups and group with high inner similarity and illustrates how this intuition in some cases could be wrong. We computed the average effectiveness (nDCG), over all the users in any group, of the group recommendations built using the five presented aggregation methods for group sizes equal to 2, 3, 4 and 8. In Figure 2 the effectiveness of the group recommendations is also compared with that of the individual recommendations. These results are based on a

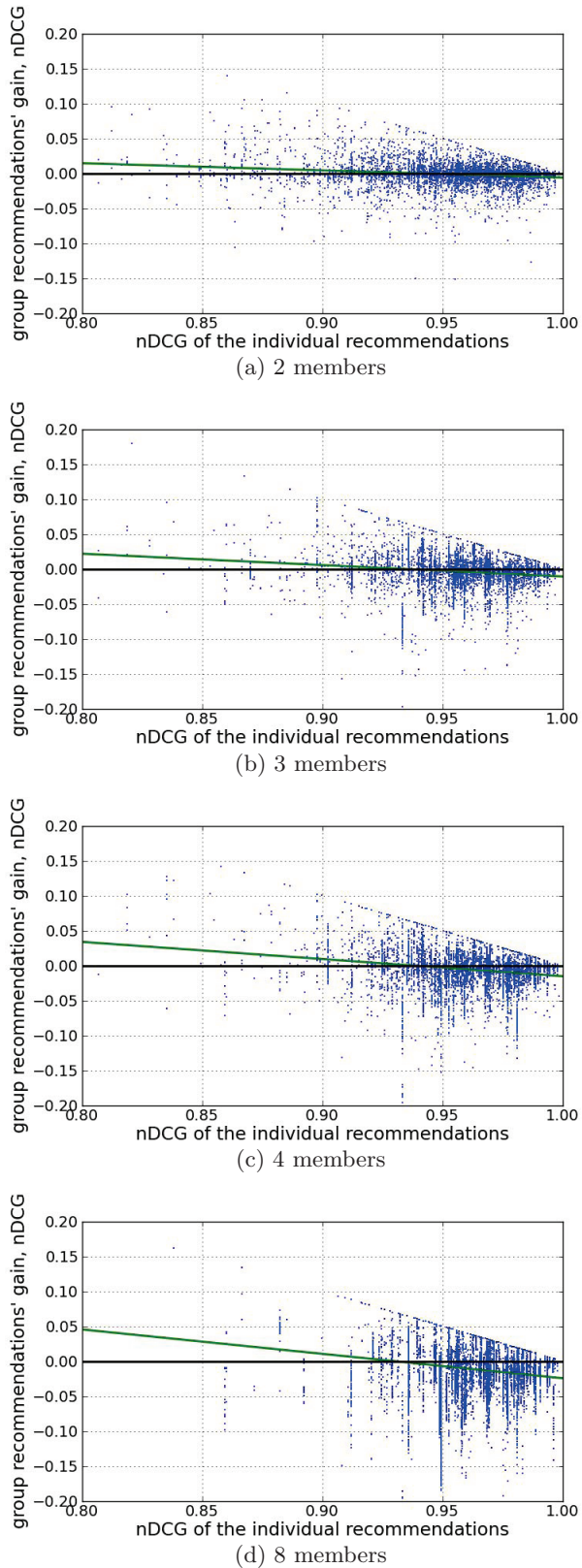


Figure 3: Gain of group recommendations with respect to individual recommendations.

sample consisting of 1000 groups for each experimental condition (e.g., 1000 random groups of size 8).

Firstly, we observe that varying the group size the variation of the effectiveness of the group recommendations is not large for groups of size 2, 3, and 4. Moreover, we see that increasing the group size the effectiveness of the group recommendations tend to decrease only for randomly generated groups. This does not hold for groups with high inner similarity; in fact, the recommendations for groups with eight members have the largest effectiveness. We note that recommendations for random groups of size 3 deviate from the general behavior: they are more effective than those for random groups of size 2. For high similarity groups the opposite holds. We do not have an explanation for that; a more careful analysis of the characteristics of the generated groups is required.

Random aggregation performs uniformly worse than any other method. The difference between random and other aggregation methods varies from 3 to 6%. Such a small variation in performance could seem surprising, however, it can be explained by the following example. Continuing the example started in Subsection 4.3, imagine to compute nDCG for the items in the user test set: $\{1, 4, 8, 7, 9\}$. Imagine that the user has given the corresponding ratings for these items: $[5, 5, 4, 3, 2]$. If the group (or personal) recommendation returns this perfect ordering, then one has a perfect match with user preferences and $nDCG = 1.0$. However, imagine, that CF rating prediction method made some mistakes and returned the ranked list $\{1, 8, 4, 9, 7\}$, with the corresponding ratings: $[5, 4, 5, 2, 3]$. nDCG of this list is still large and it is equal to 0.97. Further, if one generates a sample of random orderings of this list, on average nDCG is 0.91; which explains the high values of random aggregation in the previous experiment.

Comparing the effectiveness of group recommendations to the individual recommendations, as expected, one can observe that the individual recommendations are better ranked than the group recommendations. But this difference is very small in the groups of size 2, 3, and 4. A noticeable difference is observed only for the largest groups, i.e., with 8 members. This shows that rank aggregation is an effective approach for building group recommendations, especially for small groups, when the objective of the recommender system is to correctly rank the recommendations and not to predict the correct rating.

These results show that combining potentially conflicting rankings in a group could create a group recommendation that is not satisfactory for the group members. This happens for large groups of size 8 in both random and highly similar groups. But, even for large groups, aggregating the ranked lists predictions of the users in a group in some cases can have a positive effect. This could happen if the individual predictions are not very good. In this situation, the combination of the ranked lists of the group members can fix errors performed by the individual predictions.

This conjecture is tested in the next Section 5.2 and was originated by the observation that in many of our experimental conditions the group recommendation effectiveness is not substantially inferior than the individual one.

In conclusion, we also notice that the aggregation method itself has not a big influence on the quality, except for random aggregation. Moreover, we observe that there is no clear winner and the best performing method depends on

the group size and inner group similarity. In the following section we analyze the correlation between the effectiveness of the group and the individual recommendations, and for this goal we focus on groups of four users. This is a common group size for many activities such as movie watching with friends or traveling.

5.2 Relationship between Group and Individual Recommendations

In the second experiment we measure the difference between the effectiveness of the individual and the group recommendations' lists. We want to understand *when* the group recommendations are better or worse (ranked) than the individual recommendations. We call this difference the "gain" of effectiveness of the group recommendations. A positive gain means that the group recommendations are better ranked than the individual recommendations. As it was done in the previous experiment, we performed experiments with random groups and groups with high inner group similarity. We show results only for the average aggregation method and the high inner similarity groups. However, the results are very similar for the other settings using random groups and Borda, Spearman Footrule or Least misery aggregation methods (not shown here for lack of space).

Figure 3 shows a scatter plot where each user, in a group, is represented by a point. Here, the x axis measures nDCG for the user individual recommendation list, while the y axis shows the gain in nDCG of this group recommendation list for the same user. Note that the same user can be in several different groups, hence a user may be represented by several points. We plot only data for a sample of 3000 groups. We can see that there is a negative correlation between the gain and the effectiveness of the individual recommendations, i.e., the gain decreases as the effectiveness of the individual recommendations increases. This means that the worse (ordered) the individual recommendations are, the better (ordered) the group recommendations are, i.e., the more valuable is to aggregate the recommendation lists of the group members.

We visualized this tendency by plotting the best fit lines through the points in the scatter plots. The correlation of the fit is significant ($p < 0.001$). Specifically, it is -0.18 for groups of size 2, -0.17 of size 3, -0.19 of size 4, -0.18 of size 8 for groups with high inner group similarity and average aggregation method. We also observe that the best fit line is getting steeper when groups sizes increases.

From the analysis of this experiment we can conclude that the worse are the individual recommendations the better are the group recommendations built aggregating the recommendations for the users in the group. This is an interesting result as it shows that in real RSs, which always make some errors, some users may be better served with the group recommendations than with recommendations personalized for the individual user. When the algorithm cannot make accurate recommendations personalized for a single user it could be worth recommending a ranked list of items that is generated using aggregated information from similar users.

This result motivated the following experiment where we look at the correlation between the average similarity of a user to the other group members and the effectiveness of the group recommendations for this user. In other words we measured the effectiveness of the group recommendations for a user as a function of the user's similarity to their

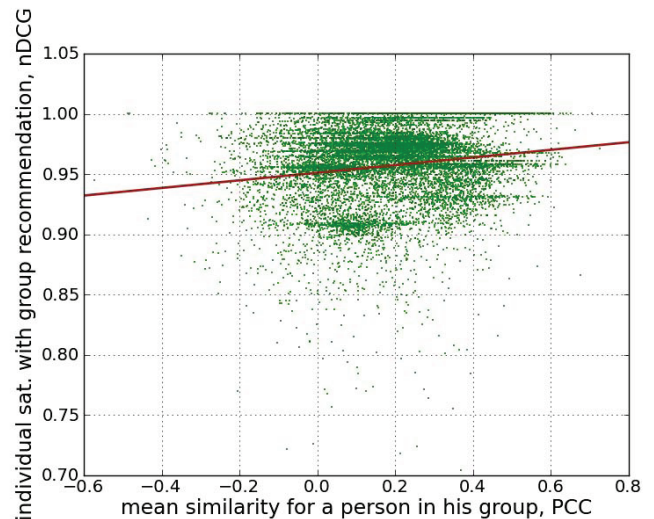


Figure 4: Correlation of the effectiveness of group recommendations with the average similarity of the user with the other group members.

peers. Here we use random groups since they better sample the possible user-to-user similarities that can be observed in groups.

Figure 4 shows the results of this experiment together with the best fit line. On the x axis we plot the average similarity of the user to all the other members of his group and on the y axis the effectiveness of the group recommendation for that user. First of all, we notice that there is a positive=0.22 (statistically significant $p < 0.001$) correlation between the similarity to the other group members and the user satisfaction. This shows, that when a user is more similar to the members of the group the group recommendation is better ordered. We observe that this correlation is not as strong as we expected; we believe this topic deserves further analysis.

6. DISCUSSION AND CONCLUSIONS

In this paper we analyzed the effectiveness of ranked list recommendations tailored for a group of users. We propose to use rank aggregation techniques for generating group recommendations from individual recommendations. We show that these recommendations are only slightly less effective than the individual recommendations for groups of moderate size (2, 3, and 4). Only for larger groups, of size 8, the group recommendation is significantly inferior than the individual recommendation.

Moreover, our results revealed some novel facts. First, we observed that the effectiveness of group recommendations does not necessarily decrease when the group size grows. In fact, this is not happening for groups of similar users. Secondly, when the individual recommendations are not correctly ranked, then recommending items ordered for a group recommendation can improve the effectiveness of the recommendations. This indicates that when, for some reason, the individual recommendations are not good, aggregating the ranked list recommendations built for a group of users, which the target user belongs to, can increase recommendation goodness. Third, we have experimentally confirmed a

common belief that the more alike are the users in the group, the more satisfied they are with the group recommendations.

In the future, we want to test our findings with other data sets, with other recommendation algorithms, and with real users. In such a way we want to further validate our findings, which are currently based on synthetically generated groups and may also be dependent on the particular collaborative filtering technique that we used, namely SVD.

In particular we would like to assess the impact in our group recommendation techniques of collaborative filtering prediction approaches aimed at producing better rankings [19] rather more accurate rating predictions. We also want to further investigate when it is convenient to make a group recommendation instead of an individual recommendation. We consider this approach as a semi-personalized recommendation strategy. With that respect it would be important to find critical user or group features that can help to decide when it is beneficial to make an individual, or a group, or even a non-personalized recommendation.

7. REFERENCES

- [1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17:734–749, 2005.
- [2] K. J. Arrow. A difficulty in the concept of social welfare. In *Journal of Political Economy* 58(4), pages 328–346. The University of Chicago Press, 1950.
- [3] S. Berkovsky and J. Freyne. Group-base recipe recommendations: analysis of data aggregation strategies. In *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*, New York, NY, USA, 2010. ACM.
- [4] R. Burke. Hybrid web recommender systems. In *The Adaptive Web*, pages 377–408. Springer Berlin / Heidelberg, 2007.
- [5] D. Coppersmith, L. Fleischer, and A. Rudra. Ordering by weighted number of wins gives a good ranking for weighted tournaments. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 776–782, New York, NY, USA, 2006. ACM.
- [6] A. Crossen, J. Budzik, and K. J. Hammond. Flytrap: intelligent group music recommendation. In *IUI '02: Proceedings of the 7th international conference on Intelligent user interfaces*, pages 184–185, New York, NY, USA, 2002. ACM.
- [7] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 613–622, New York, NY, USA, 2001. ACM.
- [8] A. Jameson and B. Smyth. Recommendation to groups. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 596–627. Springer, 2007.
- [9] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, New York, NY, USA, 2008. ACM.
- [10] C. Manning. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, 2008.
- [11] 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.
- [12] J. Masthoff and A. Gatt. In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. *User Modeling User-Adapted Interaction*, 16(3-4):281–319, 2006.
- [13] J. F. McCarthy. Pocket restaurantfinder: A situated recommender system for groups. In *In Proceedings of the Workshop on Mobile Ad-Hoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems*. Minneapolis, 2002.
- [14] J. F. McCarthy and T. D. Anagnost. Musicfx: an arbiter of group preferences for computer supported collaborative workouts. In *CSCW '98: Proceedings of the 1998 ACM conference on Computer supported cooperative work*, pages 363–372, New York, NY, USA, 1998. ACM.
- [15] K. McCarthy, M. Salamó, L. Coyle, L. McGinty, B. Smyth, and P. Nixon. Cats: A synchronous approach to collaborative group recommendation. In *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference, Melbourne Beach, Florida, USA, May 11-13, 2006*, pages 86–91, 2006.
- [16] 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.
- [17] S. Pizzutilo, B. De Carolis, G. Cozzolongo, and F. Ambruso. Group modeling in a public space: methods, techniques, experiences. In *AIC'05: Proceedings of the 5th WSEAS International Conference on Applied Informatics and Communications*, pages 175–180, Stevens Point, Wisconsin, USA, 2005. World Scientific and Engineering Academy and Society (WSEAS).
- [18] P. Resnick and H. R. Varian. Recommender systems. *Commun. ACM*, 40(3):56–58, 1997.
- [19] M. Weimer, A. Karatzoglou, Q. V. Le, and A. J. Smola. Cofi rank - maximum margin matrix factorization for collaborative ranking. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *NIPS*. MIT Press, 2007.
- [20] H. P. Young and A. Levenglick. A consistent extension of condorcet's election principle. *SIAM Journal on Applied Mathematics*, 35(2):285–300, 1978.
- [21] Z. Yu, X. Zhou, Y. Hao, and J. Gu. Tv program recommendation for multiple viewers based on user profile merging. *User Modeling and User-Adapted Interaction*, 16(1):63–82, 2006.