Design and Evaluation of a Group Recommender System

Toon De Pessemier
Wica, IBBT-Ghent University
G. Crommenlaan 8 box 201
B-9050 Ghent, Belgium
Toon.DePessemier@UGent.be

Simon Dooms
Wica, IBBT-Ghent University
G. Crommenlaan 8 box 201
B-9050 Ghent, Belgium
Simon.Dooms@UGent.be

Luc Martens
Wica, IBBT-Ghent University
G. Crommenlaan 8 box 201
B-9050 Ghent, Belgium
Luc1.Martens@UGent.be

ABSTRACT

Though most recommender systems make suggestions for individual users, in many circumstances the selected items (e.g., movies) are not for personal usage but rather for consumption in group. In this paper, we present a recommender system for audio-visual content that generates suggestions for groups of people (such as families or friends) in the home environment. In this context, different group recommendation strategies are evaluated for various algorithms and sizes of the group. An offline evaluation proves the assumption that for randomly composed groups the accuracy of all recommendation algorithms decreases if the group size grows. Besides, the results show that the group recommendation strategy which produces the most accurate results is depending on the algorithm that is used for generating individual recommendations. Consequently, if an existing recommender system for individuals is extended to a recommender system for groups, the group recommendation strategy has to be chosen based on the utilized recommendation algorithm in order to maximize the efficiency of the group recommendations.

Categories and Subject Descriptors

H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces

General Terms

Algorithms, Experimentation

Keywords

Group recommender, Evaluation, Aggregation strategy

1. INTRODUCTION

Recommender systems can help users to find the most interesting products or content thereby addressing the information overload problem of (online) services. Although the majority of the currently deployed recommender systems are designed to generate personal suggestions for individual users, in many cases content is selected and consumed by groups of users rather than by individuals. Many researchers have already investigated how the current state-of-the-art recommendation algorithms can be adapted in order to generate group recommendations [6]. In literature, group recommendations have mostly been generated either by aggregating the users' individual recommendations into recommendations for the whole group (aggregating recommendations) or by aggregating the users' individual preference model into a preference model of the group (aggregating preferences) [2].

The first recommendation strategy (aggregating recommendations) generates recommendations for each individual user using a traditional recommendation algorithm. Subsequently the recommendation lists of all group members are aggregated into a group recommendation list, which (hopefully) satisfies all group members. Different methods to aggregate the recommendation lists have been proposed during the last decade. Most of them make a decision based on the algorithm's prediction value, i.e. a prediction of the user's rating score for the recommended item. The second recommendation strategy (aggregating preferences) combines the group members' preferences into a group preference model using a social value function. A social value function describes how the opinions and preferences of individuals affect the group's recommendations. In literature, different social value functions have been proposed to aggregate the members' preferences, but still no consensus exists about the most optimal solution [5]. After aggregating the members' preferences, the group's preference model is treated as a pseudo user to produce group recommendations using a traditional recommendation algorithm.

The goal of this research is to find which of these group recommendation strategies generates the most accurate recommendations for audio-visual content in a home environment. Section 2 provides an overview of related work. Section 3 gives some information about the use case in which the results will be applied, i.e. a group recommender system for audio-visual content in the home environment. Section 4 elaborates on the implemented group recommendation strategies and Section 5 discusses the setup of our experiment. The results are presented in Section 6. Section 7 draws conclusions and points to future work.

2. RELATED WORK

In the domain of movies, Polylens is an extension of Movielens that enables recommendations for groups [6]. Polylens allows users to create and manage their own groups in order to receive group recommendations next to the traditional individual recommendations. Both survey results and observations of user behavior proved that group recommendations are valuable and desirable for the users.

A less obvious use case for group recommendations is a recipe recommender for families [2]. Since all family members typically eat a joint meal at least once a day, recipes

and food consumption are good examples of a group activity. In the context of this recipe recommender, the aggregating preferences strategy and the aggregating recommendations strategy were compared. An evaluation with a number of families showed that the aggregating preferences strategy yield slightly better results than the aggregated recommendation lists. This recommender is based on collaborative filtering (CF) and the individual data of group members is aggregated in a weighted manner, such that the weights reflect the observed interaction of group members. As was already remarked by other researchers, this is only one type of recommendation algorithm and one of the possible approaches for aggregating prediction values or recommendation lists [1]. So, an extensive comparison of the two strategies is still missing in literature. Our research compares the aggregated recommendations with the aggregating preferences strategy more thoroughly. In the context of a group recommender for movies, the two strategies are evaluated for different group sizes and different recommendation algorithms.

Research regarding the aggregating recommendations strategy has learned that the influence of the aggregation method is limited [1]. A comparison of the group recommendation lists generated using four commonly-used aggregation methods showed similar results in terms of accuracy for all methods. This study compared the results for groups with a size of 2, 3, 4 or 8 members using the aggregating recommendations strategy and an algorithm based on SVD. Our research investigates the influence of the group size not only for the strategy that aggregates the recommendations, but also for the strategy that aggregates the members' profiles. Moreover, our research also considers other sizes of the group including very large groups, and compares the results for different classes of algorithms.

3. GROUP RECOMMENDER SYSTEM

The recommender system proposed in this paper runs on a home-gateway that aggregates the content of the group members from different sources (local or remote, e.g., external hard drives, recorders, etc.) and provides an overview of their joint collection of content items (songs and videos). For each content item, a list of similar items is provided. Furthermore, personal suggestions are offered based on the preferences of the current users of the system. For scalability reasons, these suggestions are calculated by an external recommendation services and queried by the local client whenever needed. The content items and recommendations can be filtered based on genre and selected for playback on the desired device in the home environment (e.g., the television set). This interaction and viewing behavior is logged as implicit feedback for the recommender system. Besides, explicit feedback can be provided on individual items by the "thumbs up" and "thumbs down" icons or on genres, actors, and directors of the movie by selecting these attributes in the interface. Figure 1 illustrates this functionality of the recommender system with a screenshot of the user interface.

Groups can be created or changed easily by the users according to the current situation in the home. E.g., a group can be composed for the family members that are going to watch a movie this evening. In addition to adding or removing members of a group, users can assign a personal *importance weight* to each member of the group. These weights can be used to express for example that older people (such as parents) have more influence on the recommendations than younger people (such as children). Three options are possible for these weights: a high, a low, and a neutral importance. The aggregation method or social value function of the grouping strategy (Section 4) takes these importance weights into account during the calculation of the group recommendation list. Changing the group composition or the importance weights has an immediate impact on the group recommendations which are showed in interface. To enable these immediate adjustments to the recommendation list, recommendations are precalculated for every combination of group composition and importance weights. Given the small number of group members in a typical home environment and the limited options for the importance weights (3 possible values), the total number of group combinations remains limited, so that the computation load is still acceptable.

Figure 1: A screenshot of the recommender system

4. GROUP RECOMMENDATION STRATEGY

To overcome the cold start problem and evaluate the group recommendation strategies, we used the Movielens (100K) data set in the calculation process of the recommender service. Therefore, the explicit and implicit feedback provided by the users of our system will be converted to the 5-point rating scale of the Movielens system. This way, the combined data set enables the CF to find neighbors for the new users of our system and generate accurate recommendations based on the community knowledge of the Movielens data set.

Before calculating the recommendations, the user's ratings are normalized by subtracting the user's mean rating and dividing this difference by the standard deviation of the user's ratings. Some similarity metrics, such as the Pearson correlation, consider the fact that users are different with respect to how they interpret the rating scale; thereby making the normalization process unnecessary for calculating similarities. However, normalizing the ratings is still meaningful if the ratings of the group members are aggregated into a group rating before the similarities are calculated [5]. After normalization, the ratings can (optionally) be squared to incorporate the quadratic effect of feedback mechanisms. Research has proved that users may not rate items in a linear way; this means: the further away from the middle point of the scale, the larger the differences between subsequent ratings [5]. E.g., the difference between a 5-star and a 4-star rating is more significant than the difference between a 4-star and a 3-star rating.

After transforming the data, group recommendations are calculated based on the preferences of all group members. Three group recommendation strategies are available in the current implementation of the video recommender system: aggregating recommendations, aggregating preferences, or a switching strategy that combines the two previous strategies. The switching strategy generates group recommendations based on the aggregating recommendations strategy if the profile density of the group is below a certain threshold. Above that threshold, the aggregating preferences strategy is used. Switching between a strategy that aggregates the recommendations and a strategy that aggregates the preferences might produce more accurate results than both individual strategies [2].

In case of the aggregating recommendations strategy, the aggregation method calculates for each item the average of the prediction values of each group member's recommendation list. Although several alternative aggregation methods, such as "average without misery" and "least misery", are possible, research has shown that the influence of these aggregation methods on the accuracy of the group recommendations is limited [1]. In case of the aggregating preferences strategy, the members' individual preferences are aggregated into a group preference by calculating the average of the members' rating for each item. By using the same aggregation method (i.e. average) for both aggregating the individual recommendation lists and aggregating the individual preferences, the accuracy of all strategies can be compared (Section 6). If group members have an unequal importance weight, a weighted average is used as aggregation method to take the relative importance of each group member into account. Unfortunately, the influence of the importance weights on the accuracy of the group recommendations could not be evaluated in the experiment of Section 5, since the data set that was used for this research does not contain these weights.

5. EXPERIMENTAL SETUP

A number of state-of-the-art recommendation algorithms are used for comparing different group recommendation strategies. The used implementation of **Collaborative Filtering** (CF) is based on the work of Breese et al [3]. This nearest neighbor CF uses the Pearson correlation metric for discovering similar users in the user-based approach (UBCF) or similar items in the item-based approach (IBCF). As **Content-Based recommender** (CB) the InterestLMS predictor of the open source implementation of the Duine framework [7] is adopted (and extended to consider extra metadata attributes). Based on the actors, directors, and genres of the content items and the user's ratings for these items, the recommender builds a profile model for every user. The used **hybrid recommender** (Hybrid) combines the recommendations with the highest rating prediction of the IBCF and the CB recommender into a new recommendation list. The result is an alternating list of the best recommendations originating from these two algorithms. A user-centric evaluation comparing various algorithms based on various characteristics showed that this straightforward combination of CF and CB recommendations outperforms both individual algorithms on almost every qualitative metric [4]. As recommender based on matrix factorization, we opted for the open source implementation of the **SVD Recommender** (SVD) of the Apache Mahout project [8]. The recommender is configured to use 19 features, i.e. the number of genres in the Movielens data set, and the number of iterators is set at 50. To compare the results of the various recommenders, the **popular recommender** was introduced as a baseline. This recommender generates for every user always the same list of most popular items, which is based on the number of received ratings and the mean rating of each item.

A major issue in the domain of group recommender systems is the evaluation of the effectiveness. Interviewing groups or performing online evaluations can be partial solutions but are not feasible on a large scale or to extensively test alternative algorithms. Therefore, we are forced to perform an offline evaluation, in which groups are sampled from the users of a traditional single-user data set, as was done by Baltrunas et al [1]. Firstly, all users are randomly assigned to one group of a specific size. Secondly, group recommendations are generated for each of these groups . Thirdly, the recommendations are evaluated individually as in the classical single-user case, by comparing (the rankings of) the recommendations with (the rankings of) the items in the test set of the user. The evaluation of the group recommendations is based on the traditional procedure of dividing the data set chronologically in training set (60%) and test set (40%). For each user, the effectiveness of his group recommendations is evaluated based on his individual ratings in the test set using the Normalized Discounted Cumulative Gain (nDCG) [1]. We opted for a recommendation list of 5 content items, since this is a realistic length for a recommendation list in a TV interface. After calculating the nDCG for each individual user, the average nDCG over all users is calculated as an overall measure of efficiency. The group size is varying from 1 person per group (=individual recommendations) until 10 persons per group. Besides, the results are provided for very large group compositions (group sizes of 15 and 20 persons).

6. EVALUATING GROUPING STRATEGIES

Figure 2 shows the mean nDCG together with the 95% confidence intervals, according to the recommendation algorithm and the group size. Since groups are randomly created and the accuracy of the recommendations is depending on the composition of the groups, 30 measurements are performed for each combination of group size and algorithm. The average of these 30 measurements is used as an estimation of the effectiveness of the group recommendations and is visualized in Figure 2. The bar series with the prefix "Rec" are using the aggregating recommendations strategy whereas the prefix "Pref" refers to the aggregating preferences strategy. (The switching strategy is not evaluated.) The vertical axis crosses the horizontal axis at the accuracy level of the popular recommender, which is constant for the various group sizes. This way, the bar chart shows the relative improvement of each algorithm with respect to the baseline accuracy of the popular recommender. Since all combinations of group size and algorithm show an accuracy improvement with respect to the static list of most popular items, this experiment shows that group recommendations are still useful, even for large groups.

As expected, the graph shows a decreasing performance of the group recommendations as the group size increases for all algorithms. The comparison between the aggregating recommendations strategy and the aggregating preferences strategy provides another interesting finding. The grouping strategy that provides the most accurate recommendations

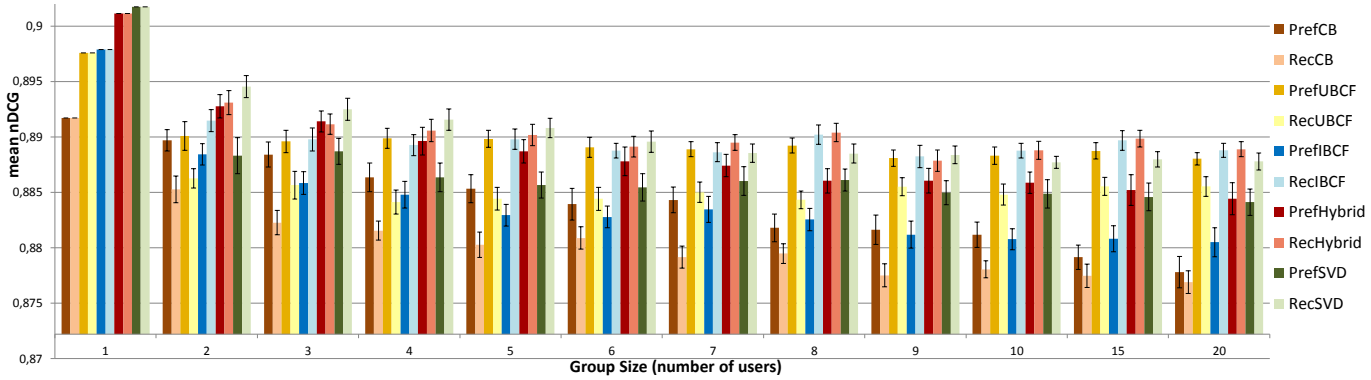

Figure 2: The accuracy of the group recommendation strategies for different algorithms and group sizes.

depends on the used algorithm. The CB and UBCF algorithm generate the most accurate group recommendations if the group members' preferences are aggregated whereas the results of SVD and IBCF are most optimal if the members' recommendations are aggregated. A possible explanation for these differences in accuracy lies in the way in which the algorithm processes the data. The CB and UBCF algorithm create some kind of user profile to find respectively matching items or similar users. In contrast, the matrix decomposition of SVD and the item-item similarities of IBCF provide less insight into the preferences of the users. So, aggregating the preferences of the group members provides optimal results if the algorithm internally composes some kind of user profile holding his preferences, whereas aggregating the recommendations of the group members is a better option if the users' preferences are less transparent in the data structure of the algorithm. The internal modeling of the user profile can also explain why some combinations of algorithm and strategy (such as PrefSVD) deteriorate faster than others (such as PrefUBCF) as the group size increases. Finally, the results of Figure 2 show that the SVD and hybrid recommender produce the most accurate group recommendations for various group sizes. However these results are only based on the Movielens data; probably the most optimal combination of algorithm and strategy depends on the data and scenario at hand. For the in-home recommender system presented in Section 3, we opted for the hybrid algorithm because of its accuracy and the positive evaluation regarding novelty, usefulness, satisfaction, and trust via previously conducted user tests [4].

7. CONCLUSIONS

We presented a group recommender for audio and video in the home environment and evaluated two commonly-used group recommendation strategies for different algorithms. Neither of these can be designated as the overall winner since the effectiveness of grouping strategies is influenced by the used recommendation algorithm. If recommender systems for individual users are extended to enable group recommendations, these results can be used to choose the most optimal grouping strategy based on the currently employed algorithm. In the future, we want to investigate the proposed switching strategy that combines the two group recommendation strategies as well as other techniques to combine the strategies. Besides, the accuracy of the grouping strategies will be compared for groups which are composed so that the group members have a high similarity.

8. REFERENCES

[1] L. Baltrunas, T. Makcinskas, and F. Ricci. Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 119–126, New York, NY, USA, 2010. ACM.

[2] S. Berkovsky and J. Freyne. Group-based recipe recommendations: analysis of data aggregation strategies. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 111–118, New York, NY, USA, 2010. ACM.

[3] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, UAI'98, pages 43–52, San Francisco, CA, USA, 1998.

[4] S. Dooms, T. De Pessemier, and L. Martens. A user-centric evaluation of recommender algorithms for an event recommendation system. In *Proceedings of the workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces at ACM Conference on Recommender Systems (RECSYS)*, 2011.

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

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

[7] Telematica Instituut/Novay. Duine Framework, 2009. Available at `http://duineframework.org/`.

[8] The Apache Software Foundation. Apache Mahout, 2012. Available at `http://mahout.apache.org/`.