Multimedia Tools and Applications manuscript No.
(will be inserted by the editor)

Comparison of group recommendation algorithms

Toon De Pessemier · Simon Dooms · Luc Martens

Received: date / Accepted: date

Abstract In recent years recommender systems have become the common tool to handle the information overload problem of educational and informative web sites, content delivery systems, and online shops. Although most recommender systems make suggestions for individual users, in many circumstances the selected items (e.g., movies) are not intended for personal usage but rather for consumption in groups.

This paper investigates how effective group recommendations for movies can be generated by combining the group members' preferences (as expressed by ratings) or by combining the group members' recommendations. These two grouping strategies, which convert traditional recommendation algorithms into group recommendation algorithms, are combined with five commonly used recommendation algorithms to calculate group recommendations for different group compositions. The group recommendations are not only assessed in terms of accuracy, but also in terms of other qualitative aspects that are important for users such as diversity, coverage, and serendipity. In addition, the paper discusses the influence of the size and composition of the group on the quality of the recommendations.

The results show that the grouping strategy which produces the most accurate results depends on the algorithm that is used for generating individual recommendations. Therefore, the paper proposes a combination of grouping strategies which outperforms each individual strategy in terms of accuracy. Besides, the results show that the accuracy of the group recommendations

T. De Pessemier - S. Dooms - L. Martens
Wica, iMinds-Ghent University
G. Crommenlaan 8 box 201, 9050 Ghent, Belgium
Tel.: +32-09-33-14908
Fax: +32-09-33-14899
E-mail: toon.depessemier@ugent.be
E-mail: simon.dooms@gent.be
E-mail: luc1.martens@ugent.be

increases as the similarity between members of the group increases. Also the diversity, coverage, and serendipity of the group recommendations are to a large extent dependent on the used grouping strategy and recommendation algorithm. Consequently for (commercial) group recommender systems, the grouping strategy and algorithm have to be chosen carefully in order to optimize the desired quality metrics of the group recommendations. The conclusions of this paper can be used as guidelines for this selection process.

Keywords group recommender · evaluation · user modeling · algorithms

1 Introduction

Recommender systems can help users to find the most interesting products or content thereby addressing the information overload problem of (online) services. Personal preferences are extracted from the users' history in order to suggest each user the most suitable items. 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. E.g., movies or TV shows are often watched in a family context, people go to restaurants, bars, and (cultural) events with their friends, and choosing a holiday destination is mostly a joint decision of the travel group. These scenarios introduce the need for discovering the most appropriate group recommendation strategies for video-on-demand services, event websites, services providing information about points-of-interest, travel agencies, etc.

The first scientific publications regarding recommender systems for groups date from the late nineties [23]. From then, many researchers have already investigated how the current state-of-the-art recommendation algorithms can be adapted in order to generate group recommendations. In the 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 models into a preference model of the group (aggregating preferences) [3]. In this paper, we refer to these strategies as *grouping strategies*.

The first grouping strategy (aggregating recommendations) generates recommendations for each individual user using a general recommendation algorithm. Subsequently, the recommendation lists of all group members are aggregated into a group recommendation list which (hopefully) satisfies all group members. Different approaches to aggregate the recommendation lists have been proposed during the last decade. Most of them make a decision based on the algorithm's prediction score, i.e. a prediction of the user's rating score for the recommended item. The higher the prediction score is, the better the match between the user's preferences and the recommended item. Aggregating the users' individual recommendations into group recommendations has some advantages. For instance, the resulting recommendations can be directly linked to the individual recommendations, which makes them easy to explain

based on the explanations of the traditional recommender [13]. Conversely, the link between the group recommendations and the individual recommendations makes it less likely to identify unexpected, surprising items [27].

The second grouping strategy (aggregating preferences) combines the users' preferences into group preferences. This way, the opinions and preferences of individual group members constitute a group preference model reflecting the interests of all members. In the literature, different approaches have been proposed to aggregate the members' preferences, but still no consensus exists about the optimal solution [21,2]. After aggregating the members' preferences, the group's preference model is treated as a pseudo user in order to produce recommendations for the group using a traditional recommendation algorithm. Compared to aggregating the individual recommendation lists, aggregating the users' preferences increases the chance of finding serendipitously valuable recommendations. On the other hand, aggregating the preferences may lead to group suggestions that lie outside the range of any individual recommendation list, which may be disorienting to the users and difficult to explain [13].

In this paper, we refer to the methods that aggregate the individual recommendation lists into group recommendations or combine the group members' preferences into a group preference model as *(data) aggregation methods*.

The remainder of this paper is organized as follows. Section 2 provides an overview of related work regarding group recommender systems. Section 3 discusses the setup of our experiment. The evaluation method is presented in Section 4. Section 5 discusses some interesting results of the experiment regarding the choice of the aggregation method and the grouping strategy. Moreover an innovative grouping strategy, combining the aggregating preferences strategy and the aggregating recommendations strategy, is proposed and evaluated. Section 6 draws conclusions and points to future work.

2 Related Work

From the late nineties, many group recommender systems have been proposed in the literature. In this section, we provide an overview of the existing group recommenders for various domains of items such as music, TV-shows and movies, touristic points-of-interest, web pages, etc.

In 1998 MusicFX was presented, a system to select background music for a group of people working out in a fitness center [23]. Based on the preferences of the people, the system constructs a group profile (by aggregating the preferences) and selects a music channel including some randomness in the choice procedure to ensure variety. According to a quantitative assessment, the vast majority of fitness center members who were involved in this trial were pleased with the group recommendations. Another music recommender for groups of users in the same environment is Flytrap [7]. Based on the music people listen to on their computers, Flytrap automatically constructs a soundtrack that tries to please everyone in the room. The system detects the presence of people in the room by the radio frequency ID badges of every user and generates

recommendations by aggregating the votes of all users (cfr. aggregating preferences strategy). Adaptive Radio is another system that selects music to play in a shared environment [5]. This recommender discovers what a user does not like instead of what the user does like. Based on these (aggregated) negative preferences, music suggestions are produced that are acceptable for all members of a group.

In the domain of movies, Polylens is an extension of MovieLens that enables recommendations for groups [27]. This recommender system uses a collaborative filtering algorithm to recommend movies for users based the users' star ratings. Polylens uses an algorithm that merges the users' recommendation lists (cfr. aggregating recommendations strategy), thereby avoiding movies that any member of the group has already rated (and therefore seen). 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. They also revealed that users are willing to share their personal recommendations with the group, thereby trading some privacy for group recommendations. In the context of recommendations for TV-content, the Family Interactive TV system (FIT) filters TV programs and creates an adaptive programming guide according to the different viewers' preferences [11]. The group recommendations of this system are based on implicit relevance feedback that is assessed through the actual program the viewer has chosen for watching. Also in the context of watching TV in group, three alternative strategies for generating group recommendations are analyzed and compared: a common group profile, aggregating recommendations, and aggregating preferences [33]. A common group profile can be considered as a virtual user of the system, representing all group members. Through a common group profile, users cannot evaluate content individually, since they have to give ratings or provide feedback for the group as a whole. The aggregating preferences strategy is chosen as optimal solution for their TV recommender. Their data aggregation method is based on total distance minimization, which guarantees that the merged result is close to most users' preferences. The evaluation results proved that the recommendation strategy is effective for multiple viewers watching TV together and appropriately reflects the preferences of the majority of the members within the group. Beside video watching in the home environment, multimedia content is often viewed by users on the move. Therefore, an adaptive vehicular multimedia system has been developed to personalize the multimedia based on the aggregation of the preferences of groups of passengers travelling together in buses, trains, and airplanes [34] (cfr. aggregating preferences strategy).

Many group recommender systems for points-of-interest (POI) such as touristic attractions, restaurants, hotels, etc. have been proposed in the literature. The Pocket Restaurant Finder provides restaurant recommendations for groups that are planning to go out eating together. The application can use the physical location of the kiosk or mobile device on which it is running, thereby taking into account the position of the people on top of their culinary

preferences. Users have to specify their preferences regarding the cuisine type, restaurant amenities, price category, and ranges of travel time from their current location on a 5-point rating scale. When a group of people is gathered together, the Pocket Restaurant Finder pools these preferences together (cfr. aggregating preferences strategy) and presents a list of potential restaurants, sorted in order of expected desirability for the group using a content-based algorithm [22]. Intrigue is a group recommender system for touristic places which considers the characteristics of subgroups such as children or disabled and addresses the possibly conflicting preferences within the group. In this system, the preferences of these heterogeneous subgroups of people are managed and combined by using a group model in order to identify solutions satisfactory for the group as a whole [1]. Also in the context of touristic activities, the Travel Decision Forum is an interactive system that assists in the decision process of a group of users planning to take a vacation together [16]. The mediator of this system directs the interactions between the users thereby helping the members of the group to agree on a single set of criteria that are to be applied in the making of a decision. This recommender takes into account people's preferences regarding various characteristics such as the facilities that are available in the hotel room, the sightseeing attractions in the surrounding area, etc [15]. An alternative recommender system for planning a vacation is CATS (Collaborative Advisory Travel System) [24]. It allows a group of users to simultaneously collaborate on choosing a skiing holiday package which satisfies the group as a whole. This system has been developed around the DiamondTouch interactive tabletop, which makes it possible to develop a group recommender that can be physically shared between up to four users. Recommendations are based on the group profile, which is a combination of individual personal preferences (cfr. aggregating preferences). The last example in the domain of POI is Group Modeller, a group recommender that provides information about museums and exhibits for small groups of people [18]. This recommender system creates group models from a set of individual user models.

Although Web browsing is usually a solitary activity, like most of today's desktop applications, various research initiatives have tried to assist a group of people in browsing by suggesting new material likely to be of common interest. Let's Browse is an extension of a single user browser that recommends web pages to a group of people using a content-based algorithm [19]. This recommender system estimates the interests of the users by analyzing the words of the visited web pages of each individual and of the groups. The system uses a simple linear combination of the profiles of each user (cfr. aggregating preferences strategy), so that the recommendation is the page that scored the best in the combined profile. Other interesting features of Let's Browse are the automatic detection of the presence of users, the dynamic display of the user profiles, and the explanation of recommendations. I-SPY is a collaborative, community-based search engine that recognizes the implicit preferences of communities of searchers and personalizes the search results [30]. This personalized search engine offers potential improvements in search performance,

especially in certain situations where communities of searchers share similar information needs and use similar queries to express these needs.

Another use case of group recommendations is a recipe recommender for families [3]. Since all family members typically eat a joint meal at least once a day, choosing a recipe and consuming the food 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 for users with low density profiles, the aggregated recommendation lists yield slightly better results than the aggregated preferences. For users with a higher density profile on the other hand, the recommendations obtained by aggregating the users' profiles showed to be more accurate, than the aggregated recommendation lists. This recommender system is based on collaborative filtering and the individual data of group members is aggregated in a weighted manner, such that the weights reflect the observed interaction of the group members. As was already remarked by other researchers, this is only one type of recommendation algorithm and one of the many possible approaches for aggregating preferences or recommendation lists [2]. So, an extensive comparison of the two grouping strategies is still missing in the literature.

Research regarding the strategy that aggregates the individual recommendation lists into a list of group recommendations (aggregating recommendations) has demonstrated that the influence of the data aggregation method is limited [2]. 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 also compared the accuracy of these group recommendations with individual recommendations (i.e. recommendations for a single user). For small groups, the group recommendations showed to be only slightly less effective than the individual recommendations, whereas for larger groups, the group recommendations are significantly inferior than the individual recommendations. If the groups are selected in such a way that the members have preferences that are quite similar, the study showed that the effectiveness of group recommendations does not necessarily decrease when the group size grows.

In this paper, we thoroughly investigate the two different strategies to generate group recommendations by comparing the accuracy of the group recommendations for various sizes of the group. Besides, the influence of the similarity between group members on the accuracy of the group recommendations is investigated. In contrast to existing research [2,3], our work goes further by comparing group recommendations generated by using various traditional recommendation algorithms. The results show that the best strategy for generating group recommendations is depending on the recommendation algorithm that is used to generate suggestions for individuals. For all algorithms, the accuracy evaluation indicates that the more alike the users of a group are, the more effective the group recommendations are. However being accurate is not enough for a recommendation list [25]; also other characteristics like diversity, coverage, and serendipity are essential for a valuable list

of suggestions. Therefore, our research also considers these additional quality metrics, whereas other studies merely focus on accuracy as the only metric for evaluating (group) recommendations [2,3].

3 Experimental Setup

3.1 Dataset

To find the best combination of group recommendation strategy and algorithm to generate suggestions for the users, the different recommendation strategies are evaluated offline using the MovieLens (100K) data set [12]. This data set contains information about 1682 popular, feature length, professionally produced movies, including 100000 evaluations on a 5-point rating scale of 943 users.

Before calculating the recommendations, the data set is first transformed to optimally estimate the preferences of the users. The user's ratings are normalized by subtracting the user's mean rating (i.e. μ) and dividing this difference by the standard deviation of the user's ratings (i.e. σ).

$$r_{norm} = \frac{r - \mu}{\sigma} \tag{1}$$

This normalization is required to compensate for very enthusiastic users giving only positive ratings or very critical users who mainly provide negative feedback. 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 [21].

3.2 Traditional Recommendation Algorithms

The focus of this research is not on developing a new group recommender from scratch but rather on investigating how effective group recommendations can be generated by combining the group members' data and using existing recommendation algorithms. Therefore, different group recommendation strategies are investigated by using a number of state-of-the-art recommendation algorithms: a content-based (CB) recommendation algorithm, a nearest neighbor collaborative filtering (CF) technique, a hybrid CF-CB algorithm (Hybrid), and a recommendation algorithm based on Singular Value Decomposition (SVD). As a baseline recommendation algorithm, we used the most-popular recommender (POP).

3.2.1 Content-Based Algorithm

Content-based recommendation algorithms generate personalized recommendations based on the metadata of the content items. As a content-based solution, the *InterestLMS predictor* of the open source implementation of the *Duine* framework [31] is adopted (and extended to consider extra metadata attributes).

Based on the metadata attributes of the content items and the user's ratings for these items, the recommender builds a profile model for every user. This profile contains an estimation of the user's preference for each genre, actor, and director that is linked to an item that the user has rated. Based on the preferences of this profile, the recommender predicts the user's preferences for unrated media items by matching the metadata of the items with the user's profile. Subsequently, the items with the highest prediction score are selected for the recommendation list.

3.2.2 Collaborative Filtering

The used implementation of collaborative filtering is based on the work of Breese et al. [4]. This nearest neighbor collaborative filter generates recommendations based on the behavior of similar users or similar items in the system. The similarity between two users or items is determined by calculating the Pearson correlation between the ratings they gave or received.

In the user-based approach (UBCF), the user's rating for an item is predicted based on the ratings of similar users. The obtained prediction score estimates how much the item will be appreciated by the user. The items with the highest prediction score are included in the recommendation list for this user. In the item-based approach (IBCF), the user's rating for an item is predicted based on his/her ratings for similar items in the system. Again, the items with the highest prediction score are recommended to this user. Experimental evaluations showed that these item-based CF algorithms are faster than the traditional user-neighborhood based recommender systems and provide recommendations with comparable or better quality [8].

3.2.3 Hybrid Recommender

The CF and CB recommender both have desired qualities, which can be combined in a Hybrid recommender. The Hybrid recommender used in this research combines the recommendations with the highest prediction score of the IBCF and the CB recommender into a new recommendation list. Because of the higher accuracy of IBCF compared to UBCF for individual recommendations, the Hybrid recommender uses the IBCF recommender as CF algorithm. The result is an alternating list of the best recommendations originating from these two algorithms (IBCF and CB). To avoid doubles, items that are recommended by the CF as well as by the CB recommender are only included once in the resulting list.

A user-centric evaluation comparing different algorithms based on various characteristics (including accuracy, novelty, diversity, satisfaction, and trust) showed that this straightforward combination of CF and CB recommendations outperforms both individual algorithms on almost every qualitative metric [9].

3.2.4 SVD

Because of their excellent performance, recommendation algorithms based on matrix factorization are commonly used. We opted for the open source implementation of the *SVD Recommender* of the *Apache Mahout* project (version 0.6) [32] in this research. The recommender is configured to use 19 factors, i.e. the number of genres in the MovieLens data set, and the number of iterations is set at 50.

3.2.5 Most-Popular Recommender

To compare the results of the different recommenders, the *most-popular recommender* was introduced as a baseline algorithm. This recommender generates for every user or group always the same static list of the most-popular items in the system, regardless the ratings or activity of the user or group. The popularity of an item is estimated by the number of ratings and the average of the ratings the item received (in the training set).

4 Evaluation Method

To find the optimal group recommendation strategy, the effectiveness of the different strategies has to be measured for various state-of-the art recommendation algorithms and different sizes of the group.

However, a major issue in the domain of group recommender systems is the evaluation of the effectiveness, i.e., comparing the generated recommendations for a group with the true preferences of the group. Performing online evaluations or interviewing groups can be partial solutions but are not feasible on a large scale or to extensively test alternative configurations. For example, in Section 5.2, five recommendation algorithms in combination with two grouping strategies are evaluated for twelve different group sizes, thereby leading to 120 different set-ups of the experiment. In addition, Section 5.3 evaluates these five algorithms and two grouping strategies for twenty additional group compositions with a varying similarity between the group members. This requires an additional number of 200 configurations. Therefore, we are forced to perform an offline evaluation, in which synthetic groups are sampled from the users of a traditional single-user data set, as was done by Baltrunas et al. [2].

In the literature, group recommendations have been evaluated several times by using a simulated data set with groups of users. Baltrunas et al. [2] used the MovieLens data set to simulate groups of different sizes (2, 3, 4, 8) and different degrees of similarity (high, random) with the aim of evaluating the

effectiveness of group recommendations. Chen et al. [6] also used the MovieLens data set and simulated groups by randomly selecting the members of the group to evaluate their proposed group recommendation algorithm. They simulated group ratings by calculating a weighted average of the group members' ratings based on the users' opinion importance parameter. Quijano-Sánchez et al. [28] used synthetically generated data to simulate groups of people in order to test the accuracy of group recommendations for movies. In addition to this offline evaluation, they conducted an experiment with real users to validate the results obtained with the synthetic groups. To measure the accuracy of the group recommendations in the online experiment, they created groups of participants and asked them to pretend that they are going to the cinema together. One of the main conclusions of their study was that it is possible to realize trustworthy experiments with synthetic data, as the online user test confirmed the results of the experiment with synthetic data. This conclusion justifies the use of an offline evaluation with synthetic groups to evaluate the group recommendations in our experiment.

The used evaluation procedure of the group recommendations, as proposed by Baltrunas et al. [2], is performed as follows. Firstly, artificial groups are composed by selecting random users from the data set. All users are assigned to one group of a pre-defined size. Secondly, group recommendations are generated for each of these groups based on the ratings of the members in the training set. Since group recommendations are intended to be consumed in group and to suit simultaneously the preferences of all members of the group, all members receive the same recommendation list. 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 in two parts: the training set, which is used as input for the algorithm to generate the recommendations, and the test set, which is used to evaluate the recommendations. In this experiment, we ordered the ratings chronologically and assigned the oldest 60% to the training set and the most recent 40% to the test set, as this reflects a realistic scenario the best. So, the ratings provided before a specific point in time are available as input for the recommender, whereas the ratings provided after that point in time are only used to evaluate the recommendations and not to train the recommender. In the remainder of this section, we discuss the quality metrics that are used to evaluate the group recommendations.

4.1 Accuracy

The accuracy of the group recommendations is evaluated based on the individual ratings in the test set using the Normalized Discounted Cumulative Gain (nDCG), a standard IR measure [20] that can be used to evaluate the recommendation lists [2].

Each recommendation list is a ranked list of n content items, $c_1, c_2, \ldots, c_n$, ordered according to their rating prediction. In this experiment, we opted for $n = 5$, since this is a realistic length for a manageable recommendation list in a TV interface. For each user, u, the accuracy of his/her group recommendations is assessed based on his/her true ratings r in the test set using the Discounted Cumulative Gain (DCG) at rank n, which is computed as:

$$DCG_n^u = r_{uc_1} + \sum_{i=2}^{n} \frac{r_{uc_i}}{log_2(i)} \tag{2}$$

Here, r_{uc_i} stands for the true rating of user u for content item c ranked in position i of the recommendation list.

The normalized DCG, nDCG, is calculated by the ratio of the DCG and the maximum DCG:

$$nDCG_n^u = \frac{DCG_n^u}{maxDCG_n^u} \tag{3}$$

where maxDCG stands for the maximum value that the DCG can get by the optimal ordering of the n content items in the recommendation list $c_1, c_2, \ldots, c_n$. The optimal ordering of the content items corresponds to the ordering of the items according to the true ratings of the user.

The calculation of the nDCG relies on the assumption that the true rating of the user is available for the recommended items. However in most cases, the test set contains only part of the items of the recommendation list. As solution to this, we adopted the suggestion of Baltrunas et al. to compute the nDCG on all the items in the test set of the user, sorted according to the ranking computed by the recommendation algorithm [2]. Using this approach, the nDCG is calculated on the projection of the recommendation list on the test set of the user. For example, suppose that $rec = [A, H, I, B, M]$ is the ordered lists of recommended items for user u, and that his/her test set contains ratings for the following seven items $test = \{Z, X, B, L, I, M, A\}$. In this case, the nDCG is computed on the ordered list $rec_{projection} = [A, I, B, M]$. After calculating the nDCG for each individual user, the average nDCG over all users is calculated as an overall measure of efficiency. This average nDCG ranges between 0 and 1; and higher values indicate more accurate group recommendations.

This accuracy evaluation, which generates synthetic groups by combining individual users, has a limitation compared to an evaluation with real groups of users. There is no way of finding out how satisfied individuals really would be with the group recommendations (in the way a real group could be asked, and real group members would take the feelings of others in the group into account). So for the offline evaluation of group recommendations based on a data set with ratings of individuals, the only possible resort is to approximate the preferences of the user being in a group, by the preferences of the user evaluating the content individually. Despite this limitation, evaluating the accuracy of group recommendations by generating synthetic groups has already proven its usefulness in previous research [2,6,28]. For the other quality metrics, such as diversity, coverage, and serendipity, the evaluation methodology based on synthetic groups is not a limitation.

4.2 Diversity

Frequently, the recommendation lists that are presented to the users contain a lot of similar items. On Amazon.com, for example, on the webpage of a book by Robert Heinlein, users receive a recommendation list full of all of his other books [25]. Indeed, recommendation algorithms can trap users in a "similarity hole", only giving exceptionally similar suggestions [25].

Accuracy metrics cannot see this problem because they are designed to judge the accuracy of the individual recommended items; they do not judge the contents of entire recommendation lists. Therefore, an additional quality metric measuring the *diversity* in the recommendation list is required. The most explored method for measuring diversity in the recommendation list uses *item-item similarity*. This item-item similarity is typically calculated based on the item content [29]. Then, the diversity of the list can be measured by calculating the sum, average, minimum, or maximum distance between item pairs. Alternatively, we could measure the value of adding each item to the recommendation list as the new item's diversity from the items already in the list [29,35].

For the use case of our recommender system for movies, it is desirable that the content items of the recommendation list are covering different genres. Therefore, we measure the item-item similarity based on the genres describing the content items. So, the item-item similarity of two content items c_i and c_j is measured by comparing the set of genres describing the first item $c_{i_{genres}}$, to the set of genres describing the second item $c_{j_{genres}}$, using the Jaccard similarity coefficient. The Jaccard similarity coefficient is a simple and effective metric which calculates the similarity of two sets by the ratio of the intersection of the sets and the union of the sets [29]:

$$Sim(c_i, c_j) = \frac{c_{i_{genres}} \cap c_{j_{genres}}}{c_{i_{genres}} \cup c_{j_{genres}}} \tag{4}$$

Subsequently, the *intra-list similarity*, i.e. a measure for the similarity of all items within a recommendation list [35], is estimated by the average of the item-item similarity of every couple of items in the list.

$$IntraList\ Similarity = \frac{2 \cdot \sum_{i=1}^{n} \sum_{j=i+1}^{n-1} Sim(c_i, c_j)}{n \cdot (n-1)} \tag{5}$$

This intra-list similarity is calculated for the recommendations of every user and the average over all users is calculated to obtain a global value for the similarity of items within a recommendation list. Finally the diversity of the recommended items is calculated by subtracting this average intra-list similarity from 1.

$$ListDiversity = 1 - average(IntraList\ Similarity) \tag{6}$$

Because of the definition of the Jaccard similarity coefficient, the average intra-list similarity ranges between 0 and 1. So, the diversity of the recommendation list varies from 0 (very similar recommendations) to 1 (very diverse recommendations). Diversity in a recommendation list is important, also in the context of group recommendations. However, it is an additional quality metric, next to accuracy, and it cannot be evaluated as a stand-alone measure, since recommendations that are more diverse, might be less accurate.

4.3 Coverage

The coverage of a recommender system is a measure of the domain of items over which the system can make recommendations [14]. In the literature, the term coverage is mainly associated with two concepts: (1) the percentage of items for which the system is able to generate a recommendation, i.e. prediction coverage, and (2) the percentage of the available items which effectively are ever recommended to a user, i.e. catalog coverage [10,14]. In this research, we focus on this second connotation of coverage, thereby providing an answer to the question: "What percentage of the available items does the recommender system recommend to users?". As a result, coverage is a metric that is especially important for the system owner and less interesting for the users. Preferably as much content items as possible are reachable through the recommendations (i.e. show up in someone's recommendation list), thereby suggesting not only the same popular items to all users, but also more niche items from the long tail matching users' specific preferences.

As suggested by Herlocker et al. [14], the catalog coverage is measured by taking the union of the top-N recommendations for each user in the population. In case the users are partitioned into groups, and group recommendations are calculated instead of individual recommendations, we measure the catalog coverage based on the union of the top-N recommendations for each of these groups. Subsequently, the cardinality of this set (i.e. the number of items in this union) is divided by the number of items in the catalog of the system to obtain the catalog coverage.

Let us denote $rec(u_i)$ as the recommendation list of user u_i. The number of users for which recommendations are generated is k and let cat be the set of all available items in the system. Then the catalog coverage can be measured as follows [10]:

$$CatalogCoverage = \frac{|\cup_{i=1\ldots k}\ rec(u_i)|}{|cat|} \tag{7}$$

The values of the catalog coverage range from 0, meaning that the recommender suggests none of the items, to 1, meaning that all items of the catalog are recommended to at least one user. Catalog coverage is usually measured on a specific set of recommendations, at a single point in time [14]. For instance in this research, it is measured based on the union of the top-5 recommendations, calculated based on the training set, for each user or group in the population.

Moreover, coverage must be measured in combination with accuracy, so recommenders are not tempted to raise coverage by making bogus predictions for every item in the system catalog [14].

4.4 Serendipity

Recommender systems might produce recommendations that are highly accurate and have reasonable diversity and coverage - and yet that are useless for practical purposes [14]. For example, a shopping cart recommender for a grocery store might suggest bananas to any shopper who has not yet selected them. Statistically, almost everyone buys bananas at the grocery store; so this recommendation is highly accurate in predicting the user's purchases. However, almost everyone who shops at the grocery store has bought bananas in the past, and knows whether or not (s)he wants to purchase more bananas. So, the shopper has already made a concrete decision whether or not to purchase bananas, and will therefore not be influenced by the recommendation for bananas. These obvious recommendations are well known to the users and do not give any new information. Much more valuable are recommendations for new products or products the customer has never heard of, but would love.

Therefore, *serendipity* is a very desirable quality attribute of a recommendation. A serendipitous recommendation helps the user find a surprisingly interesting item (s)he might not have otherwise discovered [14]. Serendipity is a measure of how surprising the successful recommendations are [29]. Like diversity and coverage, serendipity has to be balanced with accuracy, since some recommendations, such as random suggestions, might be very surprising but not relevant for the user. So, serendipity is a measure of the amount of relevant information that is new to the user in a recommendation.

Although accuracy metrics are well known and generally accepted in the domain of recommender systems, a metric for evaluating the serendipity of a recommendation list is still an open problem. Since serendipity is a measure of the degree to which the recommendations are presenting items that are both surprising and attractive to the users, designing a metric to measure serendipity is difficult [14].

Murakami et al. [26] proposed a metric for measuring the serendipity of a recommendation list by means of the concept unexpectedness. Their metric is based on the idea that the unexpectedness is low for easy-to-predict items originating from a primitive recommender and high for difficult-to-predict items coming from a more advanced recommender. Accordingly, the unexpectedness of a suggested item is estimated based on the difference between the confidence of the advanced recommender in the suggested item and the confidence of the primitive recommender in that suggested item. Unfortunately, the results obtained by this metric depend on the implementation of the primitive recommender and the resemblance between the primitive and advanced recommender. As a result, Murakami et al. introduced three possible alternatives for the primitive recommender, providing three different values for the serendip-

ity. Because of these drawbacks, we did not adopt the serendipity metric of Murakami et al. in the experiments of this paper.

Shani and Gunawardana proposed a metric for the serendipity without a dependency of a primitive recommender [29]. They proposed to estimate the serendipity by a distance measurement between a recommended content item, c_i, and the set of content items in the profile, P, of the user, i.e. the items that the user has previously watched, bought, or consumed. Although this metric is explained in the context of a book recommender and considers the authors of the books, it can easily be generalized to estimate the serendipity of any type of content item based on the metadata attributes of that item (e.g., the genres). So for the evaluation of the group recommendations, we used the following generalization of the metric of Shani and Gunawardana to estimate the serendipity of the recommended movies based on their genres (Section 5.2.4 and 5.3.4).

Let us denote $g(c_i)$ as the genre or set of genres categorizing the content item, c_i. Let $C_{p,g}$ be the number of items in the profile, P, of the user that are described by the genre, g. If g is a set of genres consisting of $\{g_1, g_2, \ldots, g_l\}$, than $C_{p,g}$ is the average of all C_{p,g_i} calculated over all genres in the set $i = 1, \ldots l$. The number of items in the user's profile that are categorized by the user's most chosen genre is represented by $C_{p,max}$:

$$C_{p,max} = \max_i(C_{p,g_i}) \tag{8}$$

The relevance of a content item, c_i, can be denoted by the boolean function $isRelevant(c_i) \in \{0, 1\}$, where $isRelevant(c_i) = 1$ means that c_i is interesting for the user, and $isRelevant(c_i) = 0$ means that it is not [26]. We consider all items in the test set that received a rating of 3, 4, or 5 stars from the user as relevant for that user. In contrast, items in the test set that received a negative rating (1 or 2 stars) are considered as uninteresting or unrelevant for the user. The personal relevance of an item that is not rated by that person is unknown and difficult to judge. Therefore, we consider unrated items as potentially relevant for the user, $isRelevant(c_i) = 1$. This favors algorithms which generate recommendations for new, unknown, or niche items, in contrast to the popular, commonly rated items. Finally, the serendipity of a recommended content item c_i can be calculated as follows:

$$Serendipity(c_i) = \frac{1 + C_{p,max} - C_{p,g(c_i)}}{1 + C_{p,max}} \cdot isRelevant(c_i) \tag{9}$$

The values of the serendipity range from 0, meaning that the recommender only suggests obvious or unrelevant items, to 1, meaning that all recommended items are relevant and surprising. Next, the *list-serendipity* is estimated by the average of the serendipity of every item in the recommendation list. The average of the list-serendipity of each user's recommendation list is used as a global measure for the serendipity of a recommendation algorithm in Section 5.2.4 and 5.3.4.

5 Results

In this section, we discuss the results of the evaluation of the group recommendations calculated by different algorithms. This evaluation is based on various quality metrics (accuracy, diversity, coverage, and serendipity) as discussed in Section 4, in order to assess the recommendations on different aspects. First, the influence of the data aggregation method on the accuracy of the group recommendations is discussed. Subsequently, this analysis evaluates the recommendations for groups of varying size and varying composition (randomly composed groups and groups with like-minded members). Finally, this section discusses how grouping strategies can be combined in order to obtain more accurate group recommendations.

5.1 Influence of the Data Aggregation Method

5.1.1 Data aggregation methods

As explained in Section 2, the (data) aggregation method is the mathematical function that determines how the individual recommendation lists of group member are combined into group recommendations in case of the aggregating recommendations strategy, or how the individual group members' preferences are combined into a group preference in case of the aggregating preferences strategy.

So, in case of the aggregating recommendations strategy, a standard recommendation algorithm (as the ones discussed in Section 3.2) is used to calculate a prediction of the user's rating for each content item in the system and for each user of the group. Next, the content items can be sorted by this prediction value in a descending order to obtain a list of recommendations for each individual user. To obtain group recommendations, the individual recommendations of the group members are aggregated by combining the prediction values of each group member's recommendation list according to the data aggregation method. Subsequently, the recommended items are sorted by this aggregated prediction value in descending order. Finally, the group recommendation list is obtained by keeping the top-N items.

In case of the aggregating preferences strategy, the members' individual preferences are aggregated into a group preference by combining the members' rating for each item according to the data aggregation method and using this aggregated result as a group rating. Subsequently, group recommendations are calculated based on these group ratings using a standard recommendation algorithm. Again, only the top-N recommendations are offered to the group.

A determining factor in the selection process of the aggregation method can be the resulting quality of the group recommendations. Therefore, the influence of the aggregation method on the accuracy of the group recommendations is investigated by comparing the following five aggregation methods, which have been proposed in the literature [21].

Average (Avg) In case of the aggregating recommendations strategy, the first aggregation method, i.e. *average*, aggregates the individual recommendation lists by calculating the average of the prediction values of the members' ratings and use this average as the prediction value for the group. In case of the aggregating preferences strategy, the average method aggregates the individual preferences by calculating the average of the members' ratings and use this average as the group rating. Because this method aggregates preferences and recommendations in a desirable and intuitive way (as discussed in Section 5.1.4), and because this method corresponds to one of the ways in which a group of people naturally make choices [21], we used this aggregation method for the experiments of Section 5.2, 5.3, and 5.4.

If group members have an unequal importance weight, which reflects the situation that some users have more influence on the group recommendations than other users, a weighted average can be 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 5.1.2, since the data set that was used for this research does not contain these weights.

Average without misery (AvgWM) The idea of the *average without misery* method is to find the optimal decision for the group, without making some group members really unhappy with this decision. If the recommendations are aggregated, the average of the prediction values of each recommendation list is calculated. Items that have a prediction value below a certain threshold (in one of the recommendation lists) get a penalty or are excluded from the group recommendations. Then the recommended items are sorted in descending order based on this new prediction value. In our implementation, the threshold is set at 2, so if an item appears in the recommendation list of a member with a prediction value of 1, the prediction value in the recommendation list of the group is set to 1. This corresponds to disfavoring the item with respect to all other available items, thereby making it very unlikely to appear in the group recommendation list.

If the preferences are aggregated, the group rating for an item is the average of the ratings of the members for that item. However, items that are rated below a certain threshold by one of the members get a penalty. Also for this strategy, the threshold is set at 2; and the penalty rule converts an individual rating below this threshold into the group rating. So if at least one group member gives a rating of 1 star to an item (i.e. below the threshold of 2 stars), the group rating is 1, otherwise the group rating is the average of the members' ratings.

One user choice (One) The aggregation method called *one user choice*, sometimes also referred to as "most respected person" or "dictatorship", adopts the preferences of one user in the group. The idea is that 1 group member might be the user that makes the decision about what the group is going to choose without consulting the other group members. In our implementation,

this user is chosen randomly from the group members. So in case of the aggregating recommendations strategy, the group's prediction value for an item is equal to the prediction value of a randomly-chosen member for that item. In case of the aggregating preferences strategy, the group's rating for an item is the rating of a randomly-chosen member for that item.

Least misery (LM) The *least misery* aggregation method tries to minimize the "misery" for the group members. The idea is that the group is as happy as its least happy member. Therefore, the goal is to obtain at least a predefined level of satisfaction for all group members. This method is implemented as follows: if the recommendations are aggregated, the group's prediction value for an item is equal to the minimum of the prediction values of all group members for that item. If preferences are aggregated, the group's rating for an item is the minimum of the members' ratings for that item.

Most pleasure (MP) The aggregation method called *most pleasure* tries to maximize the "pleasure" for (one of) the group members. This method tries to recommend alternately the items that one group member really likes, thereby not considering the preferences of other members. In case of the aggregating recommendations strategy, the group's prediction value for an item is equal to the maximum of the prediction values of all group members for that item. In case of the aggregating preferences strategy, the group's rating for an item is the maximum of the members' ratings for that item.

5.1.2 Aggregation method experiment

To investigate the influence of these data aggregation method on the accuracy of the group recommendations, group recommendations generated using each of these aggregation methods are compared via a series of experiments (Section 5.1.3). In these experiment, the groups are composed by selecting random users, meaning that no additional restrictions are imposed on the group or on the group members. To investigate the influence of the aggregation method separately from other parameters, the group size is fixed (at 2 or 5) in these experiments. For each algorithm, the two strategies to generate group recommendations (aggregating recommendations and aggregating preferences) are evaluated.

Since users are randomly combined into groups and the quality of group recommendations is depending on the composition of the groups, the quality metrics slightly vary for each partitioning of the users into groups. (Except for the partitioning of the users into groups of 1 member, which is only possible in 1 way.) Therefore, the process of composing groups by taking a random selection of users is repeated 30 times and just as much measurements of the quality metric are performed. The average of these 30 measurements is used as an estimation of the quality of the group recommendations and is visualized in the corresponding graph (Figure 1 and 2) (on the vertical axis) together with the 95% confidence intervals of the average values. The used

aggregation method is indicated on the horizontal axis. If two measurements have non-overlapping confidence intervals, they are necessarily significantly different (but if they have overlapping confidence intervals, it is not necessarily true that they are not significantly different).

The bar series with the prefix "Rec" evaluate recommendation algorithms in combination with the aggregating recommendations strategy whereas the prefix "Pref" refers to the aggregating preferences strategy. For example, the bar series "PrefUBCF" stands for the group recommendations which are generated by combining the members' individual preferences using the aggregating preferences strategy and calculating recommendations for this aggregated profile using the user-based collaborative filtering algorithm.

The vertical axes of the graphs (Figure 1 and 2) cross the horizontal axes at the quality level of the most-popular recommender (i.e. $nDCG = 0.8722$), which is constant for the different group sizes and aggregation methods. This way, the bar charts show the relative improvement (or deterioration) of each algorithm with respect to the baseline quality of the most-popular recommender.

5.1.3 Accuracy influenced by the aggregation method

Figure 1 and 2 show the average nDCG (calculated over all users) together with the 95% confidence interval of the average nDCG, in relation to the recommendation algorithm, the grouping strategy (aggregating preferences or aggregating recommendations), and the aggregation method. Figure 1 shows the accuracy of the group recommendations for groups of 2 members; whereas Figure 2 shows the accuracy for groups of 5 members.

As visible in Figure 1, the influence of the aggregation method on the accuracy of the group recommendations is largely dependent on the algorithm and grouping strategy. E.g., the accuracy of the recommendations generated by the Hybrid recommender in combination with the aggregating preferences strategy (PrefHybrid), remains approximately constant over the different aggregation methods. In contrast, the accuracy of the recommendations generated by RecCB, significantly varies if different aggregation methods are used.

The aggregation method that produces the most accurate group recommendations depends on the used algorithm and grouping strategy. E.g., the PrefCB combination produces the most accurate group recommendations if the MP method is used. If the RecCB combination is used, the most accurate group recommendations are obtained by choosing LM as aggregation method. The PrefUBCF combination provides the best results together with the AvgWM method; and the RecHybrid combination generates the most accurate recommendations if the Avg method is used. Although the confidence intervals indicate that not all differences are significant, the results show that the choice of the best aggregation method is directly linked to the grouping strategy and recommendation algorithm.

Figure 1 and 2 show that the Avg and AvgWM method generally provide the most accurate and also the most stable results. As expected, the 'one

user choice (One)' method has poor results in combination with the aggregating recommendations strategy (Rec), especially with RecCB and RecUBCF. Selecting a prediction value from one random member and ignoring the (predicted) ratings of the other members for all recommended items has a drastic influence on the resulting group recommendations. On the other hand, selecting the ratings from a random member as group rating has less influence on the final recommendations, since this happens much earlier in the recommendation process.

The LM method leads to a decreased accuracy in combination with RecUBCF and the MP method generates less accurate recommendations if RecCB or RecUBCF is used. Again, the aggregation of recommendations, which happens late in the recommendations process, can have a serious impact on the accuracy of the group recommendations because the aggregation method does not sufficiently takes into account the preferences of all members.

Comparing Figure 1 and 2 confirms that the results for groups of 2 members are in line with the results for larger groups (e.g., 5 members per group): the optimal aggregation method has to be chosen based on the used recommendation algorithm and grouping strategy. Moreover, the results of Figure 2 indicate that a sub-optimal aggregation method can have a dramatic impact on the accuracy of the recommendations, especially for larger group sizes. E.g., the accuracy of the recommendations obtained by using the aggregating recommendations strategy (Rec) and the one user choice (One) aggregation method, is significantly lower than the level of the horizontal axis, which indicates the accuracy of the list of the most popular items.

Although several other aggregation methods have been proposed in the literature [21], the results of this experiment already indicate that 'one best' aggregation method, that generates the most accurate group recommendations for all combinations of grouping strategy and algorithm, may not exist. So for an optimal group recommender system, the aggregation method has always to be chosen in combination with the recommendation algorithm and the grouping strategy.

5.1.4 Aggregation method selection

The context and application area in which group recommendations are required may also have an influence on the choice of the recommendation strategy and aggregation method. For example in a family context, meals or holiday destinations that are really disliked by one member of the family will often not be chosen for the group, regardless the opinion of the other family members. Different reasons for a strong aversion to a particular item may exist: a family member might be allergic to a specific ingredient of the meal or a family member might be (physically) unable to travel to a specific holiday destination. During these joint decisions, a solidarity between the family members exists. So, a decision that leaves one or more family members very dissatisfied is likely to be considered undesirable, even if the average satisfaction is high

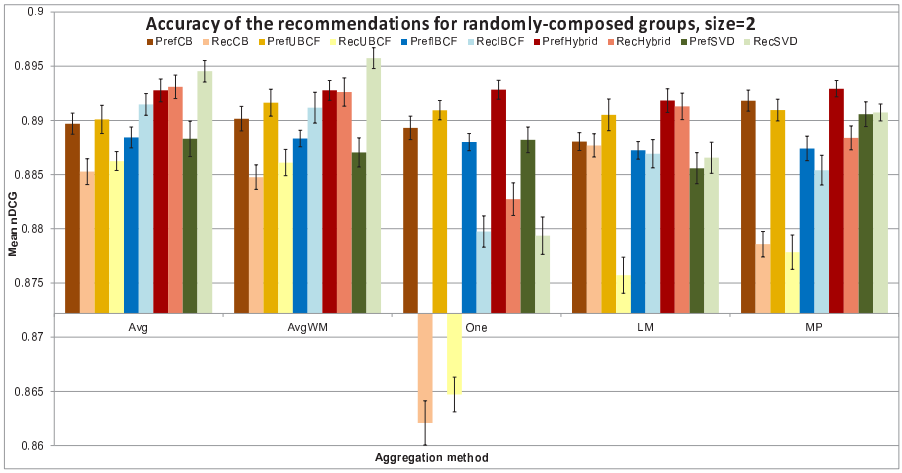

Fig. 1 The accuracy of the group recommendations for groups of size = 2, generated by using different aggregation methods

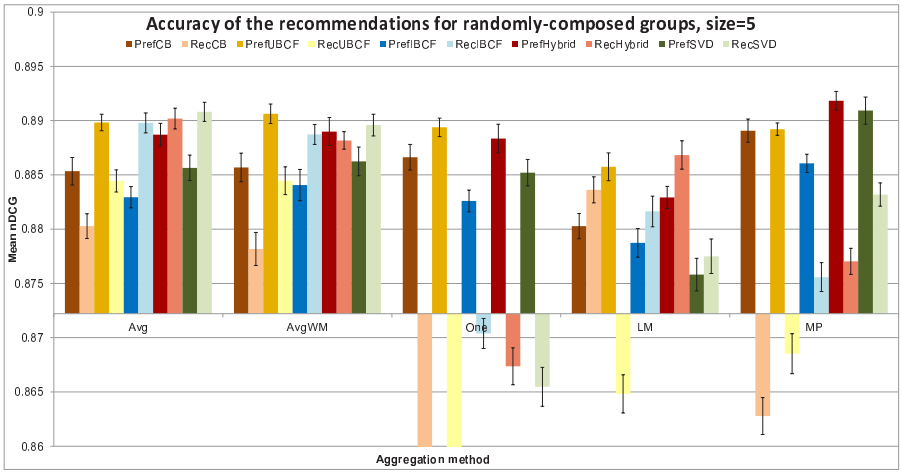

Fig. 2 The accuracy of the group recommendations for groups of size = 5, generated by using different aggregation methods

[17]. Since these items are undesirable as a group recommendation, a *minimizing misery* approach such as the *average without misery* or *least misery* aggregation method [21] is appropriate in this context.

In the context of movies or music on the other hand, users might be more willing to watch or listen to something they dislike, if the other members of the group enjoy it. E.g., people may join their friends for watching a movie or listening to music because of the company, even if they do not like some of the movies or songs during the assembly. Users might be willing to renounce their personal preferences in order to *maximize the average satisfaction* of the group. As a result, the *average* function is a proper candidate as aggregation

method. Moreover, research has shown this method to be one of the ways in which a group of people intuitively come to a group decision [21].

In this research, the different group recommendation strategies and algorithms were evaluated in the context of a recommender system for professionally produced movies that can be selected in the home environment. Because of this targeted application domain of the recommender, the *average* function was chosen in Section 5.2, 5.3, and 5.4 to combine the individual recommendation lists in the case of the aggregating recommendations strategy and to combine the members' preferences in the case of the aggregating preferences strategy. 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.

Moreover the higher average performance of the Avg method compared to the AvgWM method (Section 5.1.3) was an additional argument to chose for the Avg aggregation method for our recommender system. E.g, the recommendations for groups of 5 members generated by RecCB are significantly better in combination with the Avg method than with the AvgWM method (statistical T-test: $t(58) = 2.17, p = 0.03 < 0.05$). Consequently, all experiments of Section 5.2, 5.3, and 5.4 rely on the *average* function to aggregate preferences or recommendations.

5.2 Influence of the Group Size

The second series of experiments (Section 5.2.1, 5.2.2, 5.2.3, and 5.2.4) investigates the influence of the group size on the quality of the group recommendations. The group size is varying from 1 person per group (i.e. individual recommendations) to 10 persons per group. Besides, the results are provided for very large group compositions (group sizes of 15 and 20 persons). In contrast to the first experiments, all the combinations of grouping strategy and recommendation algorithm use the "average (Avg)" as aggregation method.

Just like in the first series of experiments, the groups are composed by selecting random users from the data set and the process of composing groups is repeated 30 times. So, each quality metric is calculated 30 times and the average of these measurements is used as an estimation of the quality of the group recommendations. The graphs in Figures 3, 4, 5, and 6 show these averages (on the vertical axis), as well as the 95% confidence intervals of the average values; the group size is indicated on the horizonal axis. Again, the vertical axis of each figure crosses the horizontal axis at the quality level of the most-popular recommender and the prefix of the bar series denotes if the algorithm uses the aggregating recommendations strategy ("Rec") or the aggregating preferences strategy ("Pref").

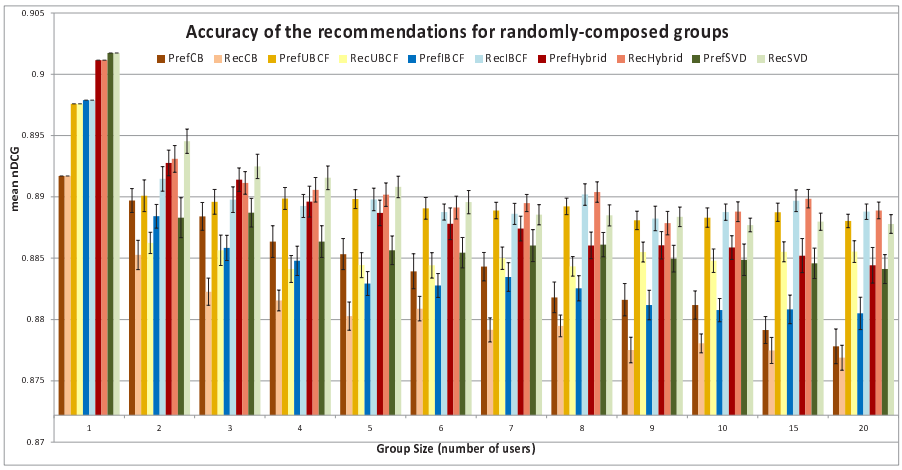

Fig. 3 The accuracy of the group recommendations for randomly-composed groups of a varying group size.

5.2.1 Accuracy Influenced by the Group Size

Figure 3 shows the average nDCG (calculated over all users) together with the 95% confidence interval of the average nDCG, in relation to the recommendation algorithm, grouping strategy, and the group size. All bar series are significantly higher than the horizontal axis indicating the accuracy level of the most-popular recommender (i.e. $nDCG = 0.8722$). So each combination of algorithm, grouping strategy, and group size shows an accuracy improvement with respect to the static list of most popular items, which demonstrates the usefulness of group recommendations, even for large groups.

A comparison of the different algorithms of Figure 3 indicates that the SVD and Hybrid recommender produce the most accurate group recommendations for various group sizes. However, the difference in accuracy with UBCF and IBCF is small. In contrast, the CB recommender generates the least accurate group recommendations, which are nevertheless still significantly better than the list of most popular items.

As expected, Figure 3 shows for all algorithms a decreasing performance regarding the accuracy of the group recommendations as the group size increases. However, this decrease is not equally large for all algorithms: a large decrease is witnessed for PrefCB, RecCB, PrefIBCF, PrefHybrid, and RecSVD, whereas PrefUBCF, RecUBCF, RecIBCF, RecHybrid, and PrefSVD suffer only from a slight decrease in accuracy as the group size increases. A larger group means more members and more individual preferences to take into account during the recommendation process. Since the groups are randomly composed, members can have different or even opposite preferences. So for these random groups, recommending items that are interesting for all members becomes more difficult when the group size increases.

Table 1 Statistical T-test comparing the accuracy obtained by the two grouping strategies for groups with size=5

Algorithm	t(58)	p-value
CB	5.03	$0.00 < 0.05$
UBCF	7.17	$0.00 < 0.05$
IBCF	-8.70	$0.00 < 0.05$
Hybrid	-1.77	$0.08 > 0.05$
SVD	-5.99	$0.00 < 0.05$

The comparison between the strategy that aggregates recommendations and the strategy that aggregates preferences provides another interesting finding. The grouping strategy that provides the most accurate recommendations 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 optimal if the members' recommendations are aggregated. The Hybrid recommender generates the most accurate recommendations in combination with the aggregating recommendations strategy, but the differences are not significant for small groups. Table 1 shows the results of the statistical T-tests comparing the accuracy of the recommendations generated by the two grouping strategies for groups of five members. (Similar results are obtained for other group sizes.) The null hypothesis, H_0 = the accuracy of the recommendations generated by the aggregating preferences strategy is equal to the accuracy of the recommendations generated by the aggregating recommendations strategy.

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 a user profile modeling the user's preferences in order to find items matching this profile (in the case of the CB algorithm) or to find users with similar preferences (in the case of UBCF). So for these algorithms, aggregating the members' preferences corresponds to aggregating the profile models of the group members. In contrast, the matrix decomposition of SVD and the item-item similarities of IBCF provide less insight into the preferences of the users or the aggregation of these preferences. The Hybrid recommender, which combines the IBCF and CB recommender, reflects the accuracy differences for the grouping strategies of the underlying algorithms.

So, aggregating the preferences of the group members provides optimal results if the algorithm internally composes some kind of user profile holding the users' 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. Consequently, if an existing recommender system for individuals is extended to a recommender system for groups, the grouping strategy has to be chosen based on the utilized recommendation algorithm in order to maximize the efficiency of the group recommendations.

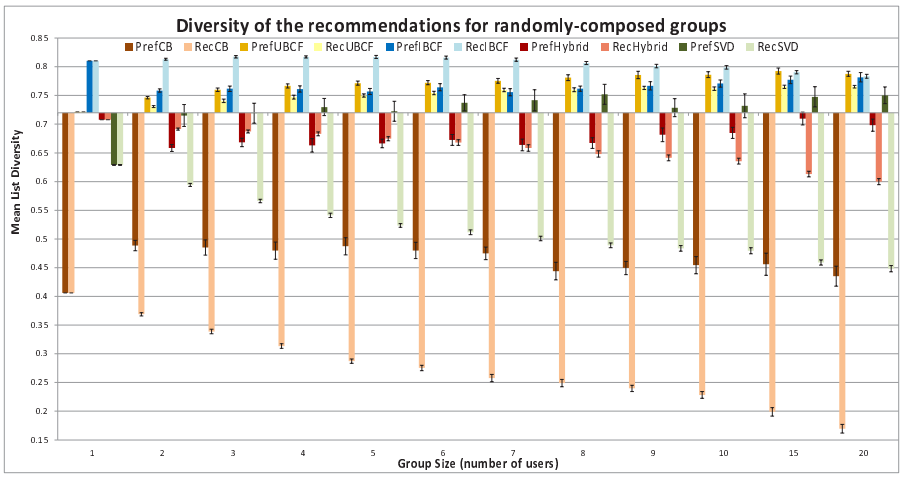

Fig. 4 The diversity of the group recommendations for randomly-composed groups of a varying group size.

5.2.2 Diversity Influenced by the Group Size

Figure 4 shows the average list diversity (calculated over all users) together with the 95% confidence interval of the average list diversity, in relation to the recommendation algorithm, grouping strategy, and the group size.

The list diversity of the most-popular recommender is 0.72, which is indicated in Figure 4 by the level of the horizontal axis. Since the most-popular recommender is based on the consumption behavior of the whole community, the suggestions consist of a set of dissimilar items covering different genres. As a result, the recommendation list generated by the most-popular recommender is rather diverse in comparison with the other algorithms such as CB and SVD.

The results reveal a clear ranking of the algorithms based on the list diversity. The CB recommender scores much worse than the most-popular recommender and produces the least diverse recommendation lists. This poor diversity is due to the reasoning process of the CB recommender. E.g., if a user gave only positive evaluations to action movies in the past, the CB recommender will only suggest more action movies to this user. In this case, the recommendation list consists of all very similar items and as a result, it has a low list diversity. This is the well-known problem of 'over-specialization' of CB recommenders. One of the purposes of hybrid systems (comparing to CB systems) is to try to overcome this problem of over-specialization. Nevertheless because of the high similarity of the CB recommendations, also the Hybrid recommender provides a recommendation list that is less diverse than the most popular list.

The recommendations based on SVD are in most cases less diverse than the most popular items. Only the recommendations based on SVD which are

generated for large groups by aggregating the members' preferences are more diverse than the most popular items. The low diversity of these recommendations might be due to the 'feature identification' of the SVD algorithm. The matrix decomposition of the algorithm reduces the user-item matrix into a smaller-dimensional space where highly correlated items (for example, movies of the same genre, same actor, ...) are captured as a single feature. Then, the resulting recommendations are characterized by the same features as the items that the user appreciated in the past.

So the CB recommender and to a lesser extent SVD can trap (individual) users in a 'similarity loop', only giving similar recommendations of the same genre over and over again, without suggesting new or surprising items to the user. If the profile of an individual user is aggregated with the profile of another user, the resulting group profile can contain a greater variety of consumed items. This is visual in the results of PrefCB and PrefSVD which show an increased list diversity when the group size grows from 1 individual user to a group of 2 members.

The algorithms based on CF generate more diverse recommendations than the most-popular recommender. The Pearson correlation metric for discovering similar users in the user-based approach (UBCF) or similar items in the item-based approach (IBCF) introduces the necessary diversity. E.g., the UBCF recommender can suggest a horror movie to a user who never rated a horror movie, because a similar user liked that horror movie. The most diverse recommendation list is obtained by using the IBCF recommender in combination with the aggregating recommendations strategy. So, the item matching process of IBCF using the Pearson correlation metric results in a very diverse set of recommendations.

For most algorithms and strategies, the diversity remains constant as the group size increases. Except for RecCB, RecSVD, and RecHybrid, the diversity decreases as the group size increases. The recommendation lists for individual users (group size=1) generated by these algorithms consist of very similar items, and combining these recommendation lists stimulates this similarity.

When we compare the two grouping strategies, SVD, UBCF and the CB recommender produce the most diverse recommendations if the preferences are aggregated whereas the group recommendations of IBCF are more diverse if the members' individual recommendation lists are aggregated. The Hybrid recommender follows the behavior of the underlying algorithms and generates more diverse recommendations for small groups if recommendations are aggregated and for large groups if preferences are aggregated. Table 2 shows the results of the statistical T-tests comparing the diversity of the recommendations generated by the two grouping strategies for groups of five members. H_0 = the diversity of the recommendations generated by the aggregating preferences strategy is equal to the diversity of the recommendations generated by the aggregating recommendations strategy.

Compared to the strategy that aggregates the recommendations, the aggregating preferences strategy combines the opinions of the different members in a very early stage of the recommendation process, thereby increasing the

Table 2 Statistical T-test comparing the diversity obtained by the two grouping strategies for groups with size=5

Algorithm	t(58)	p-value
CB	22.25	$0.00 < 0.05$
UBCF	8.06	$0.00 < 0.05$
IBCF	-17.48	$0.00 < 0.05$
Hybrid	-1.61	$0.11 > 0.05$
SVD	19.12	$0.00 < 0.05$

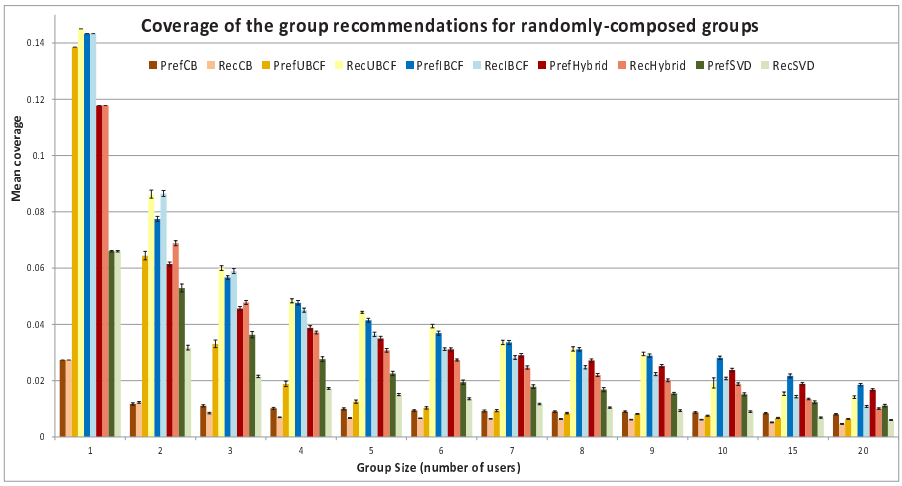

Fig. 5 The coverage of the group recommendations for randomly-composed groups of a varying group size.

diversity of the group recommendations for SVD, UBCF and CB. Combining the profiles of the different members leads to a broader group profile containing more items (SVD), which can be linked to more unconsumed items (CB), and to more neighboring users (UBCF). However since the group ratings are an average of the members' ratings, the group ratings are less extreme (i.e. closer to the middle point of the rating scale). Since the IBCF suggests the items that are most similar to the highest rated items in the profile, the recommendations based on IBCF are less diverse if the aggregating preferences strategy is used.

5.2.3 Coverage Influenced by the Group Size

Figure 5 shows the average coverage of the recommendations (calculated over all users) together with the 95% confidence interval of the average coverage, in relation to the recommendation algorithm, grouping strategy, and the group size. Since the most-popular recommender always suggests the same list of items for all users or groups regardless the preferences of the users or the size of the group, the coverage of this recommender is very low (i.e. $5/1682 = 0.00297$). Therefore, the horizontal axis crosses the vertical axis at the origin.

Table 3 Statistical T-test comparing the coverage obtained by the two grouping strategies for groups with size=5

Algorithm	t(58)	p-value
CB	12.36	$0.00 < 0.05$
UBCF	-81.64	$0.00 < 0.05$
IBCF	8.16	$0.00 < 0.05$
Hybrid	7.29	$0.00 < 0.05$
SVD	15.51	$0.00 < 0.05$

The CB recommender has the lowest catalog coverage. Because these recommendations are merely based on the metadata of the items, different groups often receive suggestions for the same items. The coverage of the recommender based on SVD is considerably higher. The recommendation lists generated by UBCF and IBCF have the least overlap for the different groups and as a result these algorithms have the highest coverage. The coverage of the Hybrid recommender is mainly due to the high coverage of the CF algorithm.

As expected, Figure 5 shows for all algorithms a decreasing coverage when the group size increases. Since all users are a member of only one group (as specified in Section 4), the number of groups decreases as the group size increases. So, more users are combined in a single group and all members of the group receive the same group recommendations. Consequently, as the group size increases, more users receive the same group recommendations and as a result the coverage decreases.

For most algorithms, the coverage obtained by using the aggregating preferences strategy is slightly higher than the coverage of the aggregated recommendations. One exception is UBCF, which has a higher catalog coverage in combination with the aggregating recommendations strategy than with the aggregating preferences strategy. Table 3 shows the results of the statistical T-tests comparing the coverage of the recommendations generated by the two grouping strategies for groups of five members. H_0 = the coverage of the recommendations generated by the aggregating preferences strategy is equal to the coverage of the recommendations generated by the aggregating recommendations strategy.

5.2.4 Serendipity Influenced by the Group Size

Figure 6 shows the average serendipity of the recommendations (calculated over all users) together with the 95% confidence interval of the average serendipity, in relation to the recommendation algorithm, grouping strategy, and the group size. The serendipity value of the list of popular recommendations is 0.43, which is indicated in Figure 6 by the level of the horizontal axis. Since the popular recommendations are based on the consumption behavior of the whole community, this recommendation list might contain items that seem surprising to some users. E.g., the list can contain movies of a genre that the user has never watched before. So in general, the list of most popular items is rather serendipitous for the users.

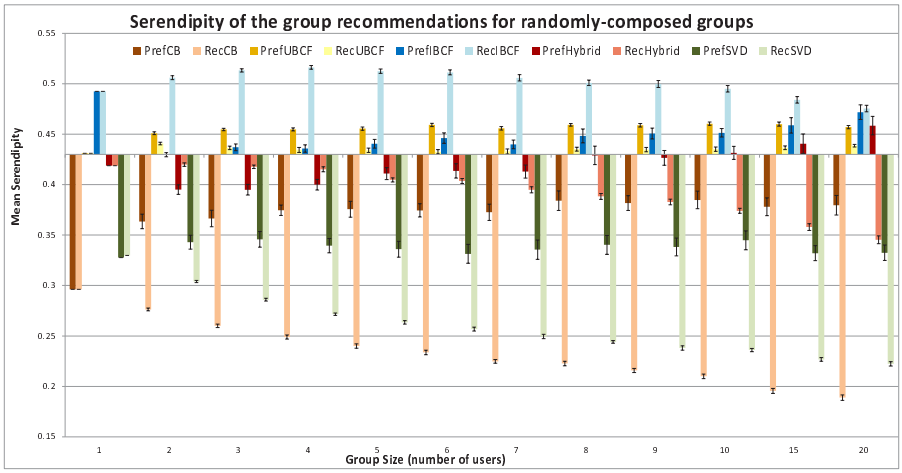

Fig. 6 The serendipity of the group recommendations for randomly-composed groups of a varying group size.

In contrast, the recommendation lists of the SVD and CB recommender contain items that users may expect. These recommenders mainly suggest items of the same genres as the items in the profile of the user, thereby not surprising the user. Consequently, the serendipity of the SVD and CB recommender is significantly lower than the serendipity of the most-popular recommender. Also the Hybrid recommender suffers from these 'too obvious' recommendations of the CB recommender. On the other hand, algorithms based on CF have the potential for serendipitous recommendations, which might be more interesting, surprising, and useful for the users.

The serendipity of most algorithms' recommendations remains constant as the group size increases. As with the diversity of the recommendations, RecCB, RecSVD, and RecHybrid are the only exceptions, showing a decreased serendipity as the group size increases.

Comparing the two grouping strategies shows that SVD, UBCF, and the CB recommender produce the most serendipitous recommendations if the preferences are aggregated whereas the group recommendations of IBCF are more serendipitous if the members' individual recommendation lists are aggregated. For the Hybrid recommender, the grouping strategy that leads to the most serendipitous recommendations is depending on the group size. Table 4 shows the results of the statistical T-tests comparing the serendipity of the recommendations generated by the two grouping strategies for groups of five members. H_0 = the serendipity of the recommendations generated by the aggregating preferences strategy is equal to the serendipity of the recommendations generated by the aggregating recommendations strategy.

Table 4 Statistical T-test comparing the serendipity obtained by the two grouping strategies for groups with size=5

Algorithm	t(58)	p-value
CB	28.59	$0.00 < 0.05$
UBCF	13.72	$0.00 < 0.05$
IBCF	-25.18	$0.00 < 0.05$
Hybrid	1.69	$0.10 > 0.05$
SVD	15.31	$0.00 < 0.05$

5.3 Influence of the Intra-group Similarity

The third series of experiments (Section 5.3.1, 5.3.2, 5.3.3, 5.3.4) investigates the influence of the similarity of group members on the quality of the group recommendations. In this series of experiments, the groups are composed of users which are more or less similar to each other.

For each measurement, the groups are created as follows. First a *minimum intra-group similarity* is determined. This is a minimum threshold for the similarity of each couple of members in the group. So each couple of users of the same group needs to have a user-user similarity that is equal to or greater than this minimum intra-group similarity. These user-user similarities are calculated by using the Pearson correlation metric on the users' ratings in the data set.

Then, groups are composed by selecting users who fulfill the requirement of the minimum intra-group similarity. The first member of the group is randomly selected without any requirement; the second member is randomly selected from the subset of users who are sufficiently similar to the first user. So the second user has a user-user similarity with the first user which is at least the defined minimum intra-group similarity. The third member of the group is randomly selected from the subset of users who are sufficiently similar to the first and the second user. This process of adding similar users to the group is repeated until the intended group size is reached. Each user can be selected for only one group, in which (s)he meets the requirement of the intra-group similarity. The result is a group of users in which every user is similar to every other user of the group with a minimum similarity as defined by the minimum intra-group similarity.

To investigate the influence of the intra-group similarity separately, the group size is fixed in these experiments whereas the minimum intra-group similarity is varying from -1.00 to 0.80 if the group size is 2 and from -1.00, to 0.55 if the group size is 5. Only the results for groups of 2 members (in Figures 7, 9, 10, and 11) and 5 members (in Figure 8) are included in this paper, since the graphs for other group sizes result in similar findings.

The minimum intra-group similarity starts at -1.00, i.e. the lowest similarity value that can be obtained by using the Pearson correlation metric. This minimum intra-group similarity of -1.00 denotes that all users are a candidate to be combined into a group. Group members can have similar preferences but they can also have completely opposite preferences. This situation corresponds

to the random group composition of Section 5.2 in which no restrictions are imposed on the group.

Further, the quality of the group recommendations is evaluated for groups with a minimum intra-group similarity of −0.75, −0.50 and −0.25. This means that the members can still have conflicting preferences but users who are complete opposites of each other (similarity of −1.00) are not allowed in the same group. Groups with a minimum intra-group similarity of 0.00 consist only of users with non-conflicting preferences; i.e. the user-user similarity of each couple of members is always positive. From then on, the recommendations are evaluated for groups with a minimum intra-group similarity that varies in steps of 0.05. As the minimum intra-group similarity increases, the condition for a user to join a group is becoming stricter. Group members have to be more similar to each other and the group becomes a homogeneous set of like-minded users.

For a group size of 2, the process of combining more similar users is stopped at a minimum intra-group similarity of 0.80. For higher values of the minimum intra-group similarity, it is not possible anymore to find a sufficient number of groups in which all users are so similar to each other. For groups of 5 users, it is even more difficult to find members who are all very similar to each other. Therefore, the minimum intra-group similarity is increased until 0.55 is reached.

Given the random aspect in the group composition (i.e. selecting a random user from the subset of users who are sufficiently similar to the other group members), the process of composing groups is repeated 30 times. Similar to the procedure of the first and second series of experiments, each metric is calculated 30 times and the average of these measurements is used as an estimation of the quality of the group recommendations.

So the graphs in Figures 7, 8, 9, 10, and 11 show these averages, as well as the 95% confidence intervals of the average values. Again, the vertical axis of each figure crosses the horizontal axis at the quality level of the most-popular recommender and the prefix of the bar series denotes if the algorithm uses the aggregating recommendations strategy ("Rec") or the aggregating preferences strategy ("Pref"). Also in these experiments the "average" function is used as aggregation method to combine the individual preferences or recommendation lists.

5.3.1 Accuracy Influenced by the Intra-group Similarity

Figure 7 shows the average nDCG (calculated over all users) for groups of two members together with the 95% confidence interval of the average nDCG, in relation to the recommendation algorithm, grouping strategy, and the minimum intra-group similarity. In this graph, two horizontal lines are indicating the accuracy of recommendations that are calculated for individual users. The green line (bottom line) represents the accuracy of recommendations calculated by the CB algorithm; this recommender has the lowest accuracy score for individual users. The red line (upper line) indicates the highest accuracy

level that was obtained for individual recommendations; these recommendations are generated using SVD.

As was already discovered by Baltrunas et al. [2], the accuracy of the group recommendations increases as the similarity between members of the group increases. The more similar the members of the group, the higher the accuracy of the group recommendations. This accuracy difference is especially noticeable for groups with a high intra-group similarity. If the minimum intra-group similarity is 0.60, the recommendations for groups of two members generated by UBCF are about as accurate as the most accurate recommendations for individuals (generated using SVD). For higher values of the minimum intra-group similarity, the accuracy of the group recommendations can transcend the accuracy level of recommendations for individuals. For example, if the minimum intra-group similarity is 0.80, all algorithms, except for the CB recommender, generate group recommendations that have a higher accuracy than the most-accurate recommendations for individuals.

This effect is even more pronounced for larger groups. Figure 8 shows the average nDCG (calculated over all users) for groups of five members together with the 95% confidence interval of the average nDCG, in relation to the recommendation algorithm, grouping strategy, and the minimum intra-group similarity. In comparison with the results of Figure 7, the accuracy of the recommendations for groups of five members is increasing faster as the minimum intra-group similarity increases. As soon as the minimum intra-group similarity is 0.25, the accuracy level of recommendations for individuals is reached. For groups of very similar users, the group recommendations of all algorithms show a significantly increased accuracy, thereby outperforming the recommendations for individuals. So if similar users are brought together in groups, even the least accurate algorithm (CB) can generate group recommendations that are more effective than the best recommendations calculated for each individual seperately.

Important to keep in mind is the fact that Figure 7 and 8 show the average nDCG for each value of the minimum intra-group similarity. So for some users the recommendations based on their individual preferences are most accurate, whereas for other users their group recommendations based on the preferences of all group members' are most accurate.

If groups are randomly composed, group members may have different or even conflicting preferences. Group recommenders have then the challenging task to generate suggestions that please all group members. Since it is not always possible to find items perfectly matching the tastes of all members, the accuracy of the group recommendations might be lower than the accuracy of the recommendations based on the individual preferences.

In contrast, if groups are composed of users with similar preferences, group recommenders do not have to deal with conflicting preferences and items that match each user's tastes can easily be found. Moreover, the group members are complementary to each other and can learn from each other's experiences with previously consumed content. If group members are similar, they will often have a comparable rating behavior. As a results, one member's ratings can

Table 5 Statistical T-test comparing the accuracy obtained for groups with a minimum intra-group similarity of -1.0 and 0.5 for groups with size=2

Algorithm	t(58)	p-value
PrefCB	-0.62	$0.53 > 0.05$
PrefUBCF	-9.64	$0.00 < 0.05$
PrefIBCF	-8.74	$0.00 < 0.05$
PrefHybrid	-7.76	$0.00 < 0.05$
PrefSVD	-5.27	$0.00 < 0.05$
RecCB	-3.35	$0.00 < 0.05$
RecUBCF	-11.33	$0.00 < 0.05$
RecIBCF	-4.83	$0.00 < 0.05$
RecHybrid	-5.13	$0.00 < 0.05$
RecSVD	-5.38	$0.00 < 0.05$

enrich the profile of another member since the ratings of both users are highly correlated. The more similar the members, the better they can complement each other, resulting in more accurate recommendations, as shown in Figure 7 and 8. Table 5 confirms this by the results of the statistical T-tests comparing the accuracy of the recommendations for groups of two members (size=2) with a minimum intra-group similarity of -1.0 and 0.5. (Similar results are obtained for other group sizes.) H_0 = the accuracy of the recommendations generated for groups with a minimum intra-group similarity of -1.0 is equal to the accuracy of the recommendations generated for groups with a minimum intra-group similarity of 0.5.

So compared to randomly-composed groups, a significant accuracy improvement is obtained for all algorithms (except for PrefCB this improvement was not significant) when the group members are similar to each other. Since the accuracy gain obtained by the similarity of group members is varying for each group, the standard deviation of the accuracy slightly increases as the minimum intra-group similarity increases. This is indicated by the size of the confidence intervals in Figure 7 and 8.

Besides the similarity of the group members, the size of the group has also an influence on the accuracy. The comparison of Figure 7 and 8 shows that if groups are randomly composed (minimum intra-group similarity of -1.00), group recommendations are most accurate for small groups. In contrast, if members are similar to each other, larger groups (Figure 8) can lead to more accurate group recommendations than smaller groups (Figure 7). E.g., the recommendations for a group of five members with a minimum intra-group similarity of 0.50 have a significantly higher accuracy than the recommendations for a group of two members with the same minimum intra-group similarity. The more users in a group, the more information and preferences that can be shared among group members. So, if these group members are similar to each other, larger groups can result in more accurate group recommendations.

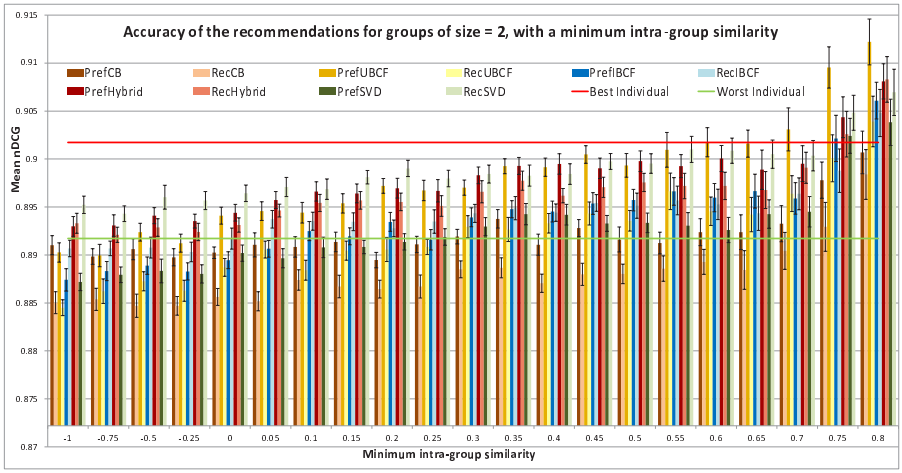

Fig. 7 The accuracy of the group recommendations for groups of size = 2, with a minimum intra-group similarity.

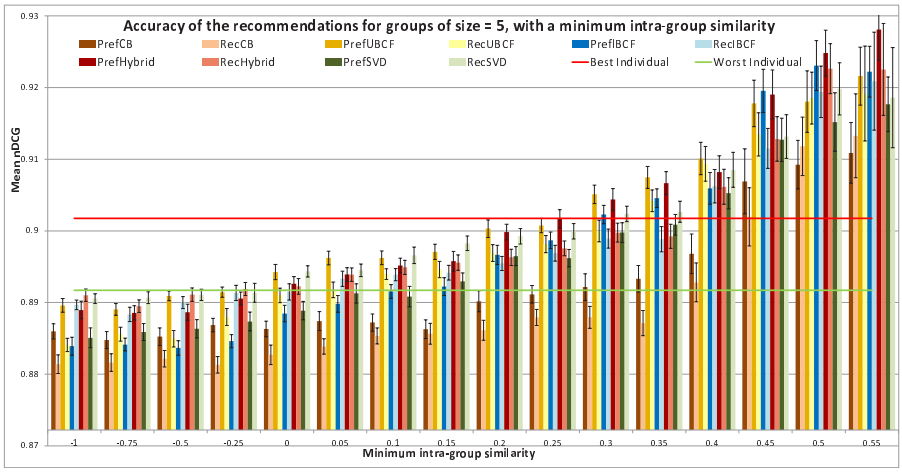

Fig. 8 The accuracy of the group recommendations for groups of size = 5, with a minimum intra-group similarity.

5.3.2 Diversity Influenced by the Intra-group Similarity

Figure 9 shows the average list diversity (calculated over all users) for groups of two members together with the 95% confidence interval of the average list diversity, in relation to the recommendation algorithm, grouping strategy, and the minimum intra-group similarity.

The results show that for PrefUBCF the list diversity slightly decreases as the minimum intra-group similarity increases. If group members are very similar to each other, all members have the same or very similar items in their profile. Aggregating these individual profiles leads to little variety in the

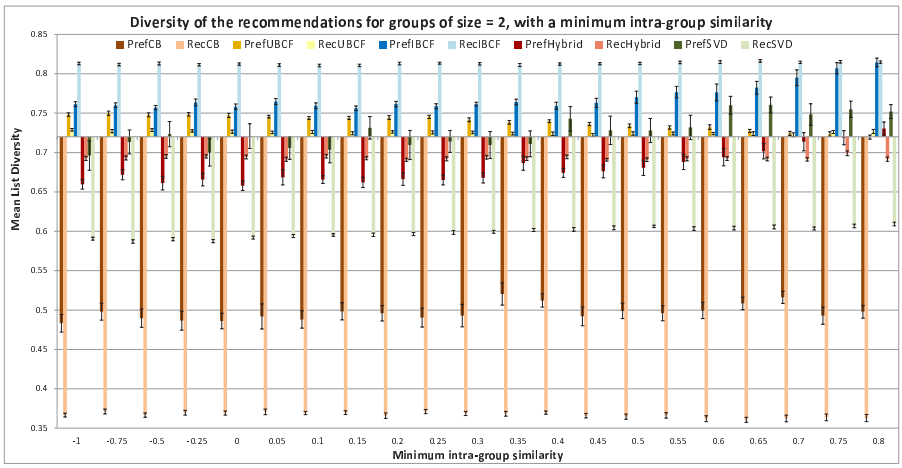

Fig. 9 The diversity of the group recommendations for groups of size = 2, with a minimum intra-group similarity.

group profile. Consequently, the recommended items are very similar to each other and so the list diversity decreases as the minimum intra-group similarity increases.

For PrefSVD and PrefIBCF on the other hand, the list diversity slightly increases as the minimum intra-group similarity increases. In contrast to UBCF, SVD and IBCF do not create a user profile modeling the user's preferences in order to generate recommendations. The increasing diversity of the PrefHybrid algorithm is due to the increasing diversity of the underling IBCF algorithm.

For the other algorithms, the list diversity remains constant as the minimum intra-group similarity increases, meaning that the similarity between group members has no influence on the list diversity. (Statistical T-test comparing the diversity obtained for groups with a minimum intra-group similarity of -1 and 0.5; group size=2; PrefUBCF: $t(58) = 13.15$, $p = 0.00 < 0.05$; PrefIBCF: $t(58) = -1.81$, $p = 0.07 > 0.05$; PrefHybrid: $t(58) = -3.13$, $p = 0.00 < 0.05$; PrefSVD: $t(58) = -2.22$, $p = 0.03 < 0.05$).

5.3.3 Coverage Influenced by the Intra-group Similarity

Figure 10 shows the average coverage of the recommendations (calculated over all users) for groups of two members together with the 95% confidence interval of the average coverage, in relation to the recommendation algorithm, grouping strategy, and the minimum intra-group similarity. The catalog coverage generally remains constant as the minimum intra-group similarity increases. So, the similarity between group members has no noteworthy influence on the catalog coverage of the group recommendations. An exception is the coverage of RecUBCF and RecIBCF that slightly decreases as the minimum intra-group similarity increases. So, these algorithms have the highest coverage for randomly-composed groups (minimum intra-group similarity = −1.00), but

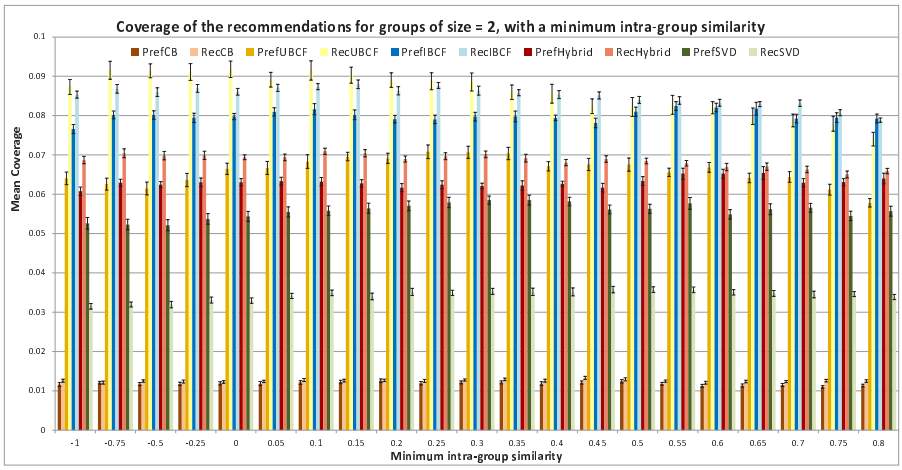

Fig. 10 The coverage of the group recommendations for groups of size = 2, with a minimum intra-group similarity.

this coverage may be slightly lower if the group members are more similar to each other. (Statistical T-test comparing the coverage obtained for the groups with a minimum intra-group similarity of -1 and 0.5; group size=2; RecUBCF: $t(58) = 4.82$, $p = 0.00 < 0.05$; RecIBCF: $t(58) = 2.21$, $p = 0.03 < 0.05$).

5.3.4 Serendipity Influenced by the Intra-group Similarity

Figure 11 shows the average serendipity of the recommendations (calculated over all users) for groups of two members together with the 95% confidence interval of the average serendipity, in relation to the recommendation algorithm, grouping strategy, and the minimum intra-group similarity. For PrefSVD and PrefIBCF the serendipity increases as the minimum intra-group similarity increases. (Statistical T-test comparing the serendipity obtained for the groups with a minimum intra-group similarity of -1 and 0.5; group size=2; PrefIBCF: $t(58) = -55.24$, $p = 0.00 < 0.05$; PrefSVD: $t(58) = -2.83$, $p = 0.01 < 0.05$). These findings are in accordance with the results for PrefSVD and PrefIBCF of Section 5.3.2, which show an increased list diversity for similar group members. So, if recommendations are more diverse, they are likely more serendipitous for the user. The serendipity of the recommendations generated by other algorithms remains constant as the minimum intra-group similarity increases.

5.4 Improved Grouping Strategy

5.4.1 Combining Strategies

The results of Section 5.2.1 showed that the used grouping strategy in combination with the recommendation algorithm has a major influence on the

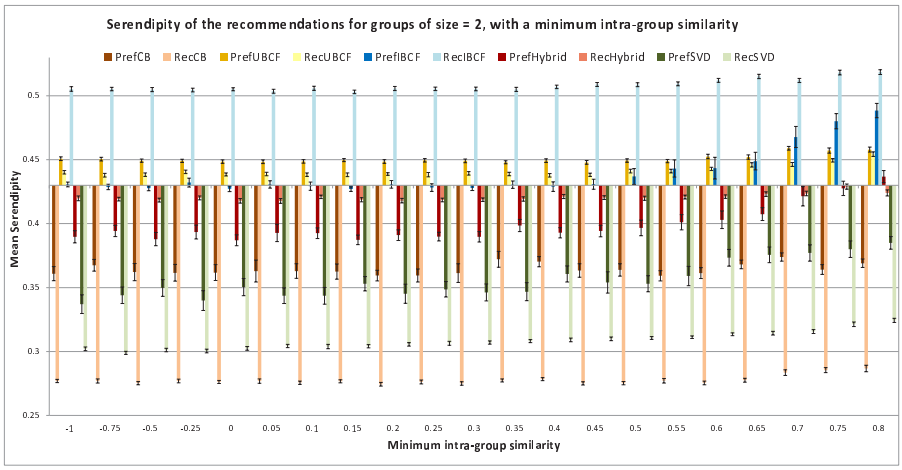

Fig. 11 The serendipity of the group recommendations for groups of size = 2, with a minimum intra-group similarity.

accuracy of the group recommendations. Certain algorithms (such as CB and UBCF) produce more accurate group recommendations when the aggregating preferences strategy is used, whereas other algorithms (such as IBCF and SVD) obtain a higher accuracy in combination with the aggregating recommendations strategy. So, the choice of the grouping strategy is crucial for each algorithm in order to obtain the best group recommendations.

Instead of selecting one individual grouping strategy, traditional grouping strategies can be combined with the aim of obtaining group recommendations which outperform the group recommendations of each individual grouping strategy. In this context, Berkovsky and Freyne [3] witnessed that the aggregating recommendations strategy outperforms the aggregating preferences strategy in terms of accuracy if the user profiles have a low density (i.e. containing a low number of consumptions). For these users, of whom little is known from their low-density profile, they obtained the lowest MAE (Mean Absolute Error for the prediction score of the group recommendations) when the aggregating recommendations strategy is used. In contrast for high-density profiles, the aggregating preferences strategy resulted in the lowest MAE, thereby outperforming the aggregating recommendations strategy in terms of accuracy. Therefore Berkovsky and Freyne proposed a switching scheme which uses the aggregating recommendations strategy in combination with a low-density profile and switches to the aggregating preferences strategy when the user profile becomes denser. Compared to the individual grouping strategies, this switching strategy yielded a small accuracy improvement.

Inspired by the proposed strategy of Berkovsky and Freyne, we employed a switching scheme that selects either the aggregating preferences or the aggregating recommendations strategy to calculate group recommendations for users of the MovieLens data set. We experimented with various switching

thresholds based on the user profile density as well as based on the group profile density. In addition, switching based on the intra-group similarity, i.e. the similarity between group members, was evaluated. However, the group recommendations obtained by using such a switching scheme did not outperform the group recommendations that are based on the best individual grouping strategy in terms of accuracy. The reason why we could not reproduce the accuracy gain of the switching scheme of Berkovsky and Freyne on the MovieLens data set might be the specific settings of their experiment. They only considered the accuracy of recommendations generated by a CF algorithm, the MAE metric was used to estimate the accuracy, and they focused on the specific use case of recipe recommendations using a rather small data set (around 3300 ratings).

Therefore, we continued our quest to a more advanced grouping strategy which combines individual grouping strategies thereby yielding an accuracy gain compared to each individual grouping strategy. The aim of this combination of strategies is to merge the knowledge of two (or more) grouping strategies into a final group recommendation list. The idea is that if one of the grouping strategies comes up with a less suitable or undesirable group recommendation, the other grouping strategy can correct this mistake. This makes the group recommendations resulting from the combination of strategies more robust than the group recommendations based on a single grouping strategy.

Although the grouping strategies can be combined in various possible ways, our experiments showed that not all techniques obtain an increased accuracy of the group recommendations. According to the results of our experiments, an effective way to generate group recommendations by combining the two grouping strategies is as follows: First, group recommendations are calculated by using the selected recommendation algorithm and the aggregating preferences strategy. The result is a list of all items, ordered according to their prediction score, which estimates how much each item will be appreciated by the group. In case of an individual grouping strategy, the top-N items on that list are selected as suggestions for the group. After calculating the group recommendations using the aggregating preferences strategy, or in parallel with it, group recommendations are generated using the chosen algorithm and the aggregating recommendations strategy. Again, the result is an ordered list of items with their corresponding prediction score.

Subsequently, the two item lists are combined into one item list by combining the prediction scores of each grouping strategy per item. In this experiment, we opted for the average as method to combine the prediction scores. So in the resulting item list, each item's prediction score is the average of the item's prediction score generated by the aggregating preferences strategy and the item's prediction score produced by the aggregating recommendations strategy. Alternative combining methods are also possible, e.g., a weighted average of the prediction scores with weights depending on the performance of each individual grouping strategy. Then the items are ordered by their new prediction score in order to obtain a new combined list of potential group recommendations.

This combined item list can still contain items that are at the top of the recommendation list that is generated by one of the grouping strategies but that are in the middle or even at the bottom of the recommendation list produced by using the other grouping strategy. Therefore, the combined item list is adapted in order to contain only items that appear at the top of both recommendation lists, thereby reducing the risk of recommending undesirable or less suitable items to the group. So, items that are ranked below a certain threshold position in the recommendation list generated by one of the grouping strategies, are removed from the combined list. In this experiment, we opted to exclude these items from the combined list, that are not in the top-5% of both recommendation lists (i.e. the top-84 items for the MovieLens data set). Since only a limited number of recommendations are offered to the users, (5 in our experiment,) the filtering of the top-5% items is no hard restriction. As a result, the final recommendation list contains the items that are identified as 'the most suitable' by both grouping strategies, ordered according to the average of the prediction scores of both grouping strategies.

Our combined grouping strategy is compared to the individual grouping strategies in Figure 12, 13, 14, and 15. Similar to the experiment of Section 5.2, the groups are composed by selecting random users from the data set and the process of composing groups is repeated 30 times. So, the graphs show the average quality metric (accuracy, diversity, coverage, or serendipity) of these 30 measurements as an estimation of the quality of the group recommendations (on the vertical axis), as well as the 95% confidence intervals of the average values. The group size is indicated on the horizonal axis. Again, the vertical axis of each figure crosses the horizontal axis at the quality level of the most-popular recommender and the prefix of the bar series denotes which grouping strategy is used. The prefix ("Combined") stands for the proposed grouping strategy which combines the aggregating preferences strategy and the aggregating recommendations strategy. The bar series with the prefix ("Best") indicates the quality level of the best individual strategy. For the individual grouping strategies, the "average (Avg)" is used to aggregate the individual preferences or recommendations.

5.4.2 Accuracy of the Combined Strategy

Figure 12 compares the accuracy of the combined grouping strategy ("Combined") and the best individual strategy ("Best"). For the UBCF and CB algorithm, the best individual strategy is the aggregating preferences strategy, whereas for the SVD, IBCF, and Hybrid algorithm the best strategy is aggregating recommendations.

The non-overlapping confidence intervals indicate a significant improvement of the combined grouping strategy compared to the best individual grouping strategy. Table 6 shows the results of the statistical T-tests comparing the accuracy of the recommendations generated by the best individual grouping strategy and the combined grouping strategy for groups with size=5. (Similar results are obtained for other group sizes.) H_0 = the accu-

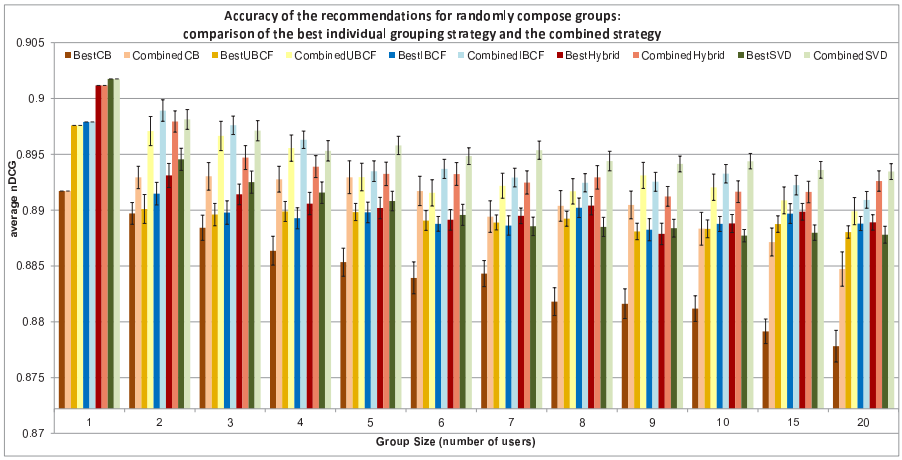

Fig. 12 The accuracy of the group recommendations for randomly-composed groups of a varying group size using the best individual grouping strategy and the combined grouping strategy.

Table 6 Statistical T-test comparing the accuracy obtained by using the best individual grouping strategy and the combined grouping strategy for groups with size=5

Algorithm	t(58)	p-value
CB	-3.55	$0.00 < 0.05$
UBCF	-2.66	$0.01 < 0.05$
IBCF	-2.33	$0.02 < 0.05$
Hybrid	-2.53	$0.01 < 0.05$
SVD	-4.39	$0.00 < 0.05$

racy of the recommendations generated by using the best individual strategy is equal to the accuracy of the recommendations generated by using the combined grouping strategy. The p-values smaller than 0.05 prove the significant accuracy improvement of our proposed grouping strategy. However, this combined grouping strategy has also a disadvantage. Since it uses the output of the individual grouping strategies, group recommendations have to be calculated for each individual strategy. As a result, the calculation load increases linearly with the number of grouping strategies that have to be combined. Fortunately, these calculations can be parallelized to speed up the total computation time.

5.4.3 Diversity of the Combined Strategy

Figure 13 compares the diversity obtained by using the combined grouping strategy and the diversity obtained by the best individual strategy. In terms of diversity, the best strategy for the CB, UBCF, and SVD recommender is aggregating preferences. In contrast, the aggregating recommendations strategy generates the most diverse recommendations for IBCF. For the Hybrid recommender, the best strategy is chosen based on the group size (aggregat-

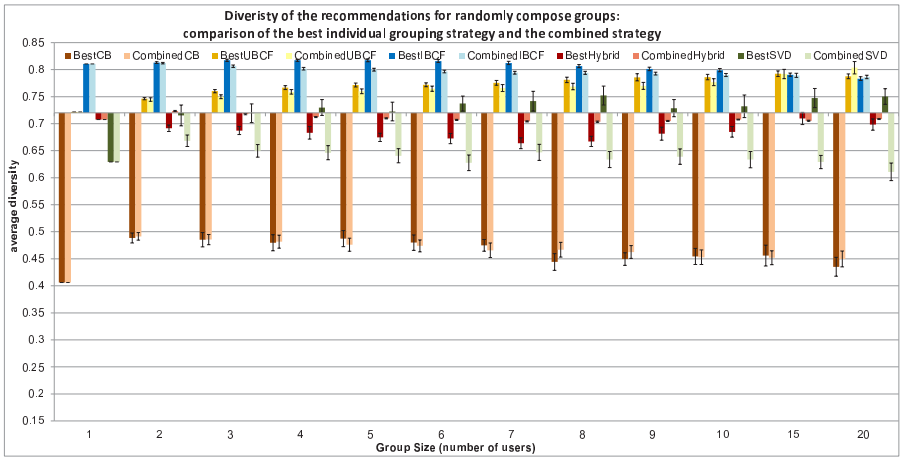

Fig. 13 The diversity of the group recommendations for randomly-composed groups of a varying group size using the best individual grouping strategy and the combined grouping strategy.

ing recommendations for a group size smaller or equal to five; aggregating preferences for larger groups).

The graph indicates that in case of the Hybrid recommender, the combined grouping strategy increases the diversity of the group recommendations, compared to the best individual grouping strategy. For the CB algorithm, the diversity of the recommendations is not significantly changed by switching from the best individual strategy to the combined grouping strategy. For the other algorithms (UBCF, IBCF, and SVD) the diversity obtained by using the combined grouping strategy is lower than the diversity of the best individual strategy. The reason for this might be the big difference in diversity between the aggregating preferences and the aggregating recommendations strategy for these algorithms, as visible in Figure 4, and the fact that the combined grouping strategy is a combination of both grouping strategies.

Table 7 shows the results of the statistical T-tests comparing the diversity of the recommendations generated by the best individual grouping strategy and the combined grouping strategy for groups with size=5. H_0 = the diversity of the recommendations generated by using the best individual strategy is equal to the diversity of the recommendations generated by using the combined grouping strategy. The T-test shows that for the CB algorithm, the diversity obtained by using the combined grouping strategy is not significantly different from the diversity obtained by using the best individual strategy (i.e. PrefCB). For the Hybrid algorithm, a significant improvement in diversity is obtained, whereas for UBCF, IBCF, and SVD a decrease in diversity is witnessed.

Table 7 Statistical T-test comparing the diversity obtained by using the best individual grouping strategy and the combined grouping strategy for groups with size=5

Algorithm	t(58)	p-value
CB	0.97	$0.33 > 0.05$
UBCF	3.01	$0.01 < 0.05$
IBCF	8.05	$0.00 < 0.05$
Hybrid	-7.94	$0.00 < 0.05$
SVD	6.19	$0.00 < 0.05$

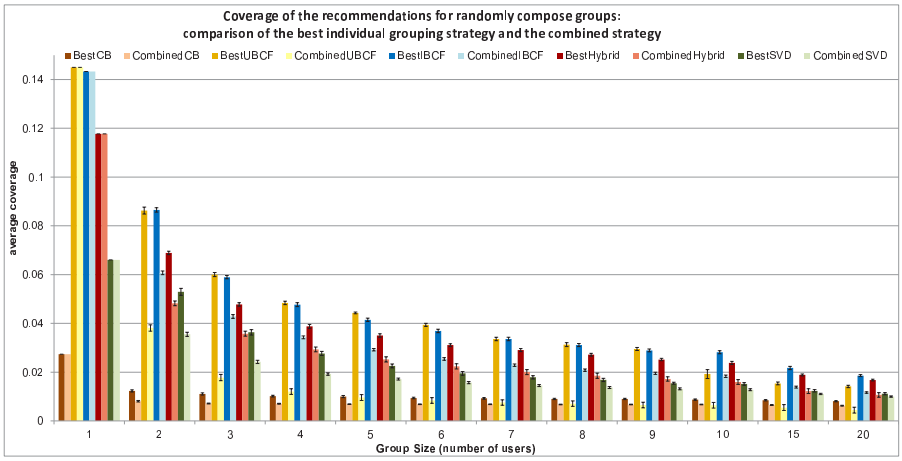

Fig. 14 The coverage of the group recommendations for randomly-composed groups of a varying group size using the best individual grouping strategy and the combined grouping strategy.

5.4.4 Coverage of the Combined Strategy

Figure 14 shows the coverage of the recommendations generated by the best individual grouping strategy and the combined grouping strategy. For the individual grouping strategy, the highest coverage is obtained by aggregating the preferences in case of the CB and SVD algorithm. In contrast, for UBCF the aggregating recommendations strategy leads to a higher coverage. For IBCF and the Hybrid recommender, the best individual grouping strategy is dependent on the group size. In comparison with the best individual strategy, the graph shows a decreased coverage for the combined grouping strategy. So, the improved accuracy of the combined grouping strategy has the side effect that different groups have a higher probability of receiving the same recommendations.

Table 8 shows the results of the statistical T-tests comparing the coverage of the recommendations generated by the best individual grouping strategy and the combined grouping strategy for groups with size=5. H_0 = the coverage of the recommendations generated by using the best individual strategy is equal to the coverage of the recommendations generated by using the combined grouping strategy. The results of the T-tests confirm the findings of Figure 14.

Table 8 Statistical T-test comparing the coverage obtained by using the best individual grouping strategy and the combined grouping strategy for groups with size=5

Algorithm	t(58)	p-value
CB	12.36	$0.00 < 0.05$
UBCF	115.95	$0.00 < 0.05$
IBCF	25.97	$0.00 < 0.05$
Hybrid	20.48	$0.00 < 0.05$
SVD	10.29	$0.00 < 0.05$

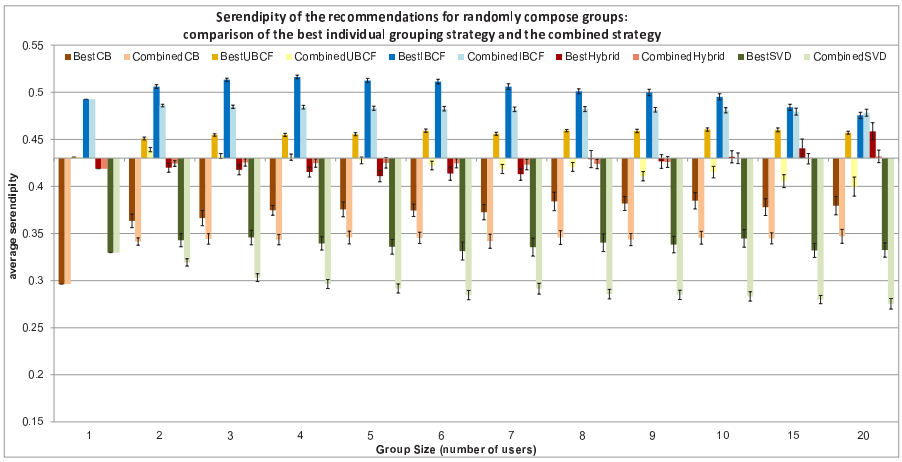

Fig. 15 The serendipity of the group recommendations for randomly-composed groups of a varying group size using the best individual grouping strategy and the combined grouping strategy.

5.4.5 Serendipity of the Combined Strategy

Figure 15 compares the serendipity of the recommendations generated by using the combined grouping strategy and by using the best individual strategy. For the CB, UBCF, and SVD algorithm, the highest serendipity is obtained by using the aggregating preferences strategy. For IBCF, the aggregating recommendations strategy leads to a higher serendipity value. For the Hybrid recommender, the best individual grouping strategy is dependent on the group size.

The graph indicates that in case of the Hybrid recommender and a group size smaller than eight, the combined grouping strategy increases the serendipity of the group recommendations, compared to the best individual grouping strategy. For the other algorithms (CB, UBCF, IBCF, and SVD) the serendipity obtained by the combined grouping strategy is lower than the serendipity of the best individual strategy. Again, the reason for this might be the big difference in serendipity between the aggregating preferences and the aggregating recommendations strategy for these algorithms, as visible in Figure 6, as well as the procedure of the combined grouping strategy, which combines both individual strategies.

Table 9 Statistical T-test comparing the serendipity obtained by using the best individual grouping strategy and the combined grouping strategy for groups with size=5

Algorithm	t(58)	p-value
CB	4.77	$0.00 < 0.05$
UBCF	16.08	$0.00 < 0.05$
IBCF	16.24	$0.00 < 0.05$
Hybrid	-3.10	$0.01 < 0.05$
SVD	8.38	$0.00 < 0.05$

Table 9 shows the results of the statistical T-tests comparing the serendipity of the recommendations generated by the best individual grouping strategy and the combined grouping strategy for groups with size=5. H_0 = the serendipity of the recommendations generated by using the best individual strategy is equal to the serendipity of the recommendations generated by using the combined grouping strategy. The T-tests indicate that for the Hybrid recommender, the serendipity obtained by the combined grouping strategy is significantly higher than the serendipity obtained by using the best individual grouping strategy. For the other algorithms, the best individual grouping strategy induces the most serendipitous recommendations, as was visible in Figure 15.

6 Conclusions

In this paper, group recommendations for movies are thoroughly evaluated in terms of multiple qualitative aspects (accuracy, diversity, coverage, and serendipity) for five state-of-the-art recommendation algorithms in combination with two commonly used grouping strategies. Furthermore, the influence of the group size and group composition on the effectiveness of the group recommendations is investigated.

The results of this paper are summarized per section in Table 10. An important result is the finding that there exists no 'overall-best' recommendation algorithm and grouping strategy. The recommendation algorithm and grouping strategy should be chosen together in order to optimize the desired qualitative aspects of the group recommendations. E.g., if the main objective of the group recommender system is to achieve a high accuracy for small to medium sized groups (size< 7), we recommend using the SVD algorithm in combination with the aggregating recommendations strategy. If other quality aspects such as diversity or coverage are also important, we recommend the IBCF or Hybrid algorithm with the aggregating recommendations strategy. When a recommender system for individual users is extended to enable group recommendations, these results can be used to choose the best grouping strategy based on the currently employed algorithm.

Future research can include the evaluation of the effectiveness of the group recommendations via an online experiment with real test subjects. In such an experiment, users can be invited to use the group recommender system at

home with their family and evaluate the group recommendations afterwards. An online experiment makes it possible to investigate if the results of the offline analysis are in line with the assessments of the users and if differences in accuracy, diversity, and serendipity are noticeable for these users.

References

1. Ardissono, L., Goy, A., Petrone, G., Segnan, M., Torasso, P.: Tailoring the recommendation of tourist information to heterogeneous user groups. In: S. Reich, M. Tzagarakis, P. De Bra (eds.) Hypermedia: Openness, Structural Awareness, and Adaptivity, *Lecture Notes in Computer Science*, vol. 2266, pp. 228–231. Springer Berlin / Heidelberg (2002)
2. Baltrunas, L., Makcinskas, T., Ricci, F.: Group recommendations with rank aggregation and collaborative filtering. In: Proceedings of the fourth ACM conference on Recommender systems, RecSys '10, pp. 119–126. ACM, New York, NY, USA (2010)
3. Berkovsky, S., Freyne, J.: Group-based recipe recommendations: analysis of data aggregation strategies. In: Proceedings of the fourth ACM conference on Recommender systems, RecSys '10, pp. 111–118. ACM, New York, NY, USA (2010). DOI http://doi.acm.org/10.1145/1864708.1864732. URL http://doi.acm.org/10.1145/1864708.1864732
4. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, UAI'98, pp. 43–52. San Francisco, CA, USA (1998). URL http://dl.acm.org/citation.cfm?id=2074094.2074100
5. Chao, D.L., Balthrop, J., Forrest, S.: Adaptive radio: achieving consensus using negative preferences. In: Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work, GROUP '05, pp. 120–123. ACM, New York, NY, USA (2005). DOI http://doi.acm.org/10.1145/1099203.1099224. URL http://doi.acm.org/10.1145/1099203.1099224
6. Chen, Y.L., Cheng, L.C., Chuang, C.N.: A group recommendation system with consideration of interactions among group members. Expert Systems with Applications **34**(3), 2082 – 2090 (2008). DOI 10.1016/j.eswa.2007.02.008. URL http://www.sciencedirect.com/science/article/pii/S0957417407000863
7. Crossen, A., Budzik, J., Hammond, K.J.: Flytrap: intelligent group music recommendation. In: Proceedings of the 7th international conference on Intelligent user interfaces, IUI '02, pp. 184–185. ACM, New York, NY, USA (2002)
8. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Trans. Inf. Syst. **22**(1), 143–177 (2004). DOI 10.1145/963770.963776. URL http://doi.acm.org/10.1145/963770.963776
9. Dooms, S., De Pessemier, T., Martens, L.: 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), pp. 67–73 (2011)
10. Ge, M., Delgado-Battenfeld, C., Jannach, D.: Beyond accuracy: evaluating recommender systems by coverage and serendipity. In: Proceedings of the fourth ACM conference on Recommender systems, RecSys '10, pp. 257–260. ACM, New York, NY, USA (2010). DOI 10.1145/1864708.1864761. URL http://doi.acm.org/10.1145/1864708.1864761
11. Goren-Bar, D., Glinansky, O.: Family stereotyping - a model to filter tv programs for multiple viewers. In: Proceedings of the 2nd Workshop on Personalization in Future TV (2002)
12. Grouplens Research: MovieLens Data Sets (2011). Http://www.grouplens.org/node/73 Accessed 13 July 2012
13. Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining collaborative filtering recommendations. In: Proceedings of the 2000 ACM conference on Computer supported cooperative work, CSCW '00, pp. 241–250. ACM, New York, NY, USA (2000)

14. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. **22**(1), 5–53 (2004). DOI 10.1145/963770.963772. URL http://doi.acm.org/10.1145/963770.963772
15. Jameson, A.: More than the sum of its members: challenges for group recommender systems. In: Proceedings of the working conference on Advanced visual interfaces, AVI '04, pp. 48–54. ACM, New York, NY, USA (2004)
16. Jameson, A., Baldes, S., Kleinbauer, T.: Two methods for enhancing mutual awareness in a group recommender system. In: Proceedings of the working conference on Advanced visual interfaces, AVI '04, pp. 447–449. ACM, New York, NY, USA (2004)
17. Jameson, A., Smyth, B.: The adaptive web. chap. Recommendation to groups, pp. 596–627. Springer-Verlag, Berlin, Heidelberg (2007). URL http://dl.acm.org/citation.cfm?id=1768197.1768221
18. Kay, J., Niu, W.: Adapting information delivery to groups of people. In: Proceedings of the Workshop on New Technologies for Personalized Information Access at the Tenth International Conference on User Modeling (2006)
19. Lieberman, H., van Dyke, N., Vivacqua, A.: Let's browse: a collaborative browsing agent. Knowledge-Based Systems **12**(8), 427 – 431 (1999). DOI 10.1016/S0950-7051(99)00036-2. URL http://www.sciencedirect.com/science/article/pii/S0950705199000362
20. Manning, C.D., Raghavan, P., Schtze, H.: Introduction to information retrieval. Cambridge Univ. Press, New York, NY (2008)
21. Masthoff, J.: Group modeling: Selecting a sequence of television items to suit a group of viewers. User Modeling and User-Adapted Interaction **14**, 37–85 (2004)
22. McCarthy, J.: Pocket restaurantfinder: A situated recommender system for groups. In: Proceedings of the Workshop on Mobile AdHoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems. ACM (2002)
23. McCarthy, J.F., Anagnost, T.D.: Musicfx: an arbiter of group preferences for computer supported collaborative workouts. In: Proceedings of the 1998 ACM conference on Computer supported cooperative work, CSCW '98, pp. 363–372. ACM, New York, NY, USA (1998)
24. McCarthy, K., Salamo, M., Coyle, L., McGinty, L., Smyth, B., Nixon, P.: Cats: A synchronous approach to collaborative group recommendation. In: G. Sutcliffe, R. Goebel (eds.) FLAIRS Conference, pp. 86–91. AAAI Press (2006)
25. McNee, S.M., Riedl, J., Konstan, J.A.: Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: CHI '06 extended abstracts on Human factors in computing systems, CHI EA '06, pp. 1097–1101. ACM, New York, NY, USA (2006). DOI 10.1145/1125451.1125659. URL http://doi.acm.org/10.1145/1125451.1125659
26. Murakami, T., Mori, K., Orihara, R.: Metrics for evaluating the serendipity of recommendation lists. In: K. Satoh, A. Inokuchi, K. Nagao, T. Kawamura (eds.) New Frontiers in Artificial Intelligence, *Lecture Notes in Computer Science*, vol. 4914, pp. 40–46. Springer Berlin / Heidelberg (2008)
27. O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J.: Polylens: a recommender system for groups of users. In: Proceedings of the seventh conference on European Conference on Computer Supported Cooperative Work, ECSCW'01, pp. 199–218. Norwell, MA, USA (2001). URL http://dl.acm.org/citation.cfm?id=1241867.1241878
28. Quijano-Sanchez, L., Recio-Garcia, J.A., Diaz-Agudo, B.: Personality and social trust in group recommendations. In: Proceedings of the 2010 22nd IEEE International Conference on Tools with Artificial Intelligence - Volume 02, ICTAI '10, pp. 121–126. IEEE Computer Society, Washington, DC, USA (2010). DOI 10.1109/ICTAI.2010.92. URL http://dx.doi.org/10.1109/ICTAI.2010.92
29. Ricci, F., Rokach, L., Shapira, B., Kantor, P.B.: Recommender Systems Handbook, 1st edn. Springer-Verlag New York, Inc., New York, NY, USA (2010)
30. Smyth, B., Balfe, E., Freyne, J., Briggs, P., Coyle, M., Boydell, O.: Exploiting query repetition and regularity in an adaptive community-based web search engine. User Modeling and User-Adapted Interaction **14**, 383–423 (2004). URL http://dx.doi.org/10.1007/s11257-004-5270-4. 10.1007/s11257-004-5270-4
31. Telematica Instituut/Novay: Duine Framework (2009). Http://duineframework.org/ Accessed 13 July 2012
32. The Apache Software Foundation: Apache Mahout (2012). Http://mahout.apache.org/ Accessed 13 July 2012

33. Yu, Z., Zhou, X., Hao, Y., Gu, J.: Tv program recommendation for multiple viewers based on user profile merging. User Modeling and User-Adapted Interaction **16**, 63–82 (2006). URL http://dl.acm.org/citation.cfm?id=1146521.1146531
34. Zhiwen, Y., Xingshe, Z., Daqing, Z.: An adaptive in-vehicle multimedia recommender for group users. In: Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st, vol. 5, pp. 2800 – 2804 Vol. 5 (2005)
35. Ziegler, C.N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: Proceedings of the 14th international conference on World Wide Web, WWW '05, pp. 22–32. ACM, New York, NY, USA (2005). DOI 10.1145/1060745.1060754. URL http://doi.acm.org/10.1145/1060745.1060754

Experiment	Section	Results
Aggregation method	5.1.3	The 'average' and 'average without misery' method generally produce the most accurate group recommendations. The 'one user choice' method induces a low accuracy in combination with the aggregating recommendations strategy.
Accuracy random groups	5.2.1	The accuracy of the recommendations decreases as the group size increases. The grouping strategy that generates the most accurate group recommendations depends on the algorithm. For CB and UBCF, aggregating preferences is the best strategy. For SVD and IBCF, the best strategy is aggregating the recommendations.
Diversity random groups	5.2.2	The CB algorithm generates the least diverse recommendation list, even less diverse than the most-popular list. Algorithms based on CF generate the most diverse recommendations. For most algorithms the diversity remains constant as the group size increases. For SVD, UBCF, and CB the aggregating preferences strategy generates the most diverse recommendations. For IBCF the aggregating recommendations strategy generates the most diverse recommendations.
Coverage random groups	5.2.3	The CB recommender has the lowest coverage. Recommenders based on CF have the highest coverage. For most algorithms (except UBCF) and group sizes, the coverage obtained using the aggregating preferences strategy is slightly higher than the coverage of the aggregated recommendations.
Serendipity random groups	5.2.4	The serendipity of the recommendations generated by the SVD and CB algorithm is significantly lower than the serendipity obtained by the most-popular recommender. Algorithms based on CF have the potential for serendipitous recommendations. The serendipity of most algorithms' recommendations remains constant as the group size increases. The SVD, UBCF, and CB recommender produce the most serendipitous recommendations if the preferences are aggregated whereas the recommendations of IBCF are most serendipitous if the members' individual recommendation lists are aggregated.
Accuracy similar groups	5.3.1	The more similar the members of the group, the higher the accuracy of the recommendations. Compared to randomly-composed groups, the group recommendations show a significantly increased accuracy for groups of similar users, with the potential of outperforming the recommendations for individuals.
Diversity similar groups	5.3.2	For most algorithms, the list diversity remains constant as the similarity between group members increases. For PrefSVD, PrefIBCF, and RecSVD on the other hand, the list diversity slightly increases as the similarity between group members increases.
Coverage similar groups	5.3.3	The coverage generally remains constant as the similarity between group members increases.
Serendipity similar groups	5.3.4	For PrefSVD and PrefIBCF (and to a lesser extent for RecSVD) the serendipity increases as the similarity between group members increases. The serendipity of the recommendations generated by other algorithms remains constant as the similarity between group members increases.
Combining Strategies	5.4	Compared to the best individual grouping strategy, a significant accuracy improvement can be obtained by combining both strategies, at the expense of a decreased diversity, coverage, and serendipity (for most algorithms).

Table 10 Conclusions of the study on group recommendations