Tampere University

Ilmo Heiska

MULTI-ROUND RECOMMENDATIONS FOR STABLE GROUPS

Faculty of Information Technology and Communication Sciences
M. Sc. Thesis
April 2021

ABSTRACT

Ilmo Heiska: Multi-Round Recommendations for Stable Groups
M.Sc. Thesis
Tampere University
Master's Degree Programme in Computer Sciences
April 2021

Recommender systems have been used for suggesting the most suitable products and services for users in diverse scenarios. More recently, the need for making recommendations for groups of users has become increasingly relevant. In addition, there are applications in which recommendations are required in a consecutive sequence. Group recommendations present a challenge for recommender systems: how to balance the preferences of the individual members of a group. On the other hand, when making recommendations for a group for multiple consecutive rounds, a recommender system has a possibility to dynamically try to balance the preference differences between the group members.

This thesis suggests two novel group recommendation methods for multi-round group recommendation scenarios: *adjusted average aggregation method* and *average-min-disagreement aggregation method*. Both of the novel methods aim to provide a group with highly relevant results for the group while remaining fair for all group members. An experimental evaluation is designed and implemented as a 15-round recommendation sequence in order to assess the performance of the two novel methods. The experiment includes several types of groups with different degree of similarity between group members, to check the performance of the methods in various different scenarios. A recently introduced recommendation method, *sequential hybrid aggregation method*, is used as a baseline method for multi-round group recommendation performance.

The experimental results show that the two novel methods exceed the performance of the baseline method in all scenarios. Of the two novel methods, adjusted average method outperforms average-min-disagreement method in the early stages of the multi-round recommendation sequence. Average-min-disagreement method, on the other hand, achieves better overall results in the later stages of the multi-round recommendation sequence. Additionally, the average-min-disagreement method achieves the most fair results for all group members in all scenarios.

Key words and terms: recommender systems, group recommendations, multi-round recommendations, fair recommendations.

The originality of this thesis has been checked using the Turnitin Originality Check service.

Contents

1. Introduction

Recommender systems are used increasingly in various application domains from music and movie recommendations to item and service recommendations on the internet. The constantly accelerating use of social media has created the need for making recommendations to groups of users in addition to more traditional single-user recommendations. The balancing of the different preferences of a group's users is a challenging requirement for any group recommender system. Additionally, if a group asks for recommendations in a consecutive sequence, the group recommendation task becomes even more challenging, if past interactions are taken into consideration. On the other hand, executing a sequence of group recommendations makes it possible for the recommender system to dynamically try to adjust and balance the satisfaction and disagreement of the individual users in the group during the sequence.

Most recommender systems provide recommendations as a list of the most relevant items for a user. There are two main approaches to producing a recommendation list: the content-based method [2] and the collaborative filtering method [3]. Content-based recommender systems recommend items to the target user that are similar to the items the user has preferred in the past. Specifically, content-based recommender systems aim to identify target user preferences by analyzing the items the user has consumed in the past. The content-based movie recommender, for example, would then suggest new movies for the target user that are most similar to the profile of the user's interests. For a content-based movie recommender the item features that are analyzed could include movie genre, actors, plot, movie duration, tags, themes, for example. Consider a movie recommendation scenario where the system has data that the target user has watched multiple movies with Tom Hanks in the cast and the most watched genres for the target user are comedy and drama. A content-based recommender system would likely then recommend such new movies to the target user that are tagged with drama and/or comedy genre as well as movies where Tom Hanks is acting.

Collaborative filtering recommender systems recommend items to the target user that other similar users have preferred. Consider a similar movie recommendation scenario as before, where the target user has given high ratings for drama and comedy movies and movies with Tom Hanks. The recommender would then recommend such movies to the target user that similar users have given high ratings for.

For single-user recommendations, a recommender system has a relatively straightforward task to recommend to the target user the items that are most relevant to the user. But increasingly, there are situations where a recommender system is required to make recommendations for multiple users that are interconnected, namely *group recommendations*. Typical scenarios requiring group recommendation are finding a good restaurant

for family dinner, a travel destination for a group of friends, a book to read for a book club, a movie to watch at a movie night for a group of friends. Making recommendations for a group of users presents a challenge compared to standard single-user recommendations as it is unlikely that the group members share exactly the same preferences in any real-life scenario. Therefore, the recommender system has to be able to balance the preferences of the individual users in the group. The problem of preference balancing can be considered from two points of view for the recommender system. First of all, the recommender system should select items to recommend to the group that are highly relevant for all group members. Secondly, at the same time, the recommender system should try to ensure that each group member is not disproportionally satisfied or unsatisfied with the group recommendation result. In other words, the recommender system should aim to have all group members evenly satisfied with the group recommendation result. This second viewpoint can be considered as minimizing the group disagreement that is the difference in the degree of satisfaction, or dissatisfaction, between group members, regarding the group recommendation result.

In practice, there are many application domains where a user interacts with the recommender system for multiple times in a row, namely *multi-round recommendations*. The standard recommendation methods have no "memory", but multi-round recommendation methods also consider the past user interactions. Scenarios where a group of users require recommendations multiple times in a sequence can be considered as *multi-round group recommendations*. Multi-round group recommendations offer a possibility to further balance the group recommendation results and to try to satisfy the group members more evenly. Such a scenario occurs when a group of friends get together weekly for a movie night, for instance. If one of the group members is more satisfied than other group members with the group recommendation result this week, a multi-round group recommender system can give more weight to the other group members' opinion when aggregating the group recommendation result the following week. Or, if one of the group members is unsatisfied with the group recommendation results this week, a multi-round group recommender system can give more weight to that user's opinion when aggregating the group recommendation result the following week.

In this thesis, two novel multi-round group recommendation methods are proposed that aim to produce high quality recommendation results with low disagreement between the group members. Both of the two proposed methods are based on collaborative filtering with the score aggregation approach to generating a group recommendation list. The two proposed methods share a common target of producing high quality results while maintaining low group disagreement but have slight differences in the overall strategy. One of the methods considers the previous interactions of the group and dynamically increases the influence of those users that have less-than-average satisfaction from the previous

round. The other method uses a two-stage process for making the final group recommendation list. In the first stage, the group recommendation list is generated using a score aggregation approach. In the second stage, the top items in the group recommendation list are reordered according to a function that minimizes the difference between the individual users' predicted ratings and the group-aggregated rating for each item.

A hybrid multi-round group recommendation method was proposed by Stratigi et al. [9] that uses a weighted combination of two typical score aggregation methods for group recommendations and dynamically adjusts the weights during a group recommendation sequence. The method aims to generate a group recommendation list for a group, at each step of the recommendation sequence, with highly relevant items without sacrificing the opinions of the minority of the group. The method was recently shown to have better overall performance compared to standard group recommendation methods [9] and therefore can be justified as a baseline method for current multi-round group recommendation method performance.

The main contributions of this thesis are the following:

- A novel multi-round group recommendation method is proposed that takes into account the previous interactions of the group with the system and adjusts the influence a group member has with the final group recommendation list.
- A novel multi-round group recommendation method is also proposed that uses a two-stage strategy for making the group recommendation list. In the first stage, the group recommendation list is generated with score aggregation and, in the second stage, the top items in the list are reordered to minimize group disagreement.
- The performance of the proposed methods is evaluated and compared with a recently introduced hybrid multi-round group recommendation model in an experimental study.

The thesis is organized as follows. Section 2 describes the related work. In Section 3, the main concepts for a multi-round group recommender system are presented. The details on the three multi-round recommendation methods that are used in the experiments are presented in Section 4. Section 5 describes the experimental setup and Section 6 presents the relevant results from the experimental evaluation. Finally, the thesis is concluded in Section 7.

2. Related work

The work most closely related to this thesis is the recent work [9] in which the authors proposed a novel multi-round group recommendation method, *sequential hybrid aggregation*, that produces highly relevant recommendations for a group without sacrificing the minority opinion in the group. From the two novel methods proposed in this thesis, *sequential adjusted average aggregation* uses a similar strategy. While sequential hybrid aggregation can only consider the opinion of one user in the minority, sequential adjusted average considers all users with less-than-average satisfaction as the minority. The other proposed method, *average-min-disagreement*, uses a different strategy but with similar targets: to produce highly relevant results with low disagreement between group members.

2.1. Recommender systems

Most recommender systems are designed to utilize user information to provide a user, or a group of users, with a list of items that are most relevant to them. There are two main approaches for this: the content-based method [2] and the collaborative filtering method [3]. Both approaches require prior knowledge of the items the target user has consumed in the past. The main difference between the two approaches is that a content-based recommender system requires more detailed knowledge of the items in the system that the user has not yet consumed, while a collaborative filtering recommender system does not require detailed knowledge of the items as such, but instead utilizes the ratings of other users in the system. Content-based recommender systems typically create a profile of the target user's interests by analyzing the features of the items the user has consumed. Then, the system recommends to the user new items that are similar to the ones that the user has preferred in the past.

Collaborative filtering recommender systems use the ratings of other users as the basis of recommendations for the user. Collaborative filtering methods can be further divided into two main approaches: model-based and memory-based collaborative filtering algorithms. [11] Model-based algorithms [12] predict ratings for the users in the system by first creating a model for the users' behavior and then using these models to predict user preferences. Memory-based algorithms [13, 14] are based on a user-ratings matrix that contains the ratings each user has given to items in the system. A recommender would first locate similar users from the user-ratings matrix who have given similar ratings to the same movies as the target user. The ratings of similar users and a prediction function are then used to predict ratings for new items for the target user. Finally, the system will recommend to the target user the top items according to the predicted rating.

2.2. Group recommendations

In many practical applications a group of users do not share identical preferences. This presents a challenge for a recommender system as it has to be able to balance the preferences of the individual users in the group. There are two main approaches to tackle this challenge, namely the so-called virtual user approach and score aggregation. [5] The virtual user approach combines the group members' preferences and generates a virtual user profile by aggregating the ratings of each group member. Then, standard single-user recommendations can be applied for that virtual user. The score aggregation approach utilizes a single-user recommendation method to generate a recommendation list for each user in the group separately and then a group recommendation list is aggregated from each user's individual recommendation list. In this thesis, the score aggregation approach is used in the proposed methods since the baseline method in the experiment, *sequential hybrid aggregation* is also based on score aggregation approach.

Average aggregation and least misery aggregation are the two main approaches of score aggregation that have been proposed. [5] The former approach computes the group recommendation list by taking the average score for each item in the group members' individual recommendation lists. The latter approach on the other hand generates the group recommendation list by taking the lowest score for each item in the group members' recommendation lists. Table 1 illustrates average and least misery aggregation methods.

Table 1. Group recommendation scores with average and least misery aggregation methods for users *A-E* and the predicted score for each user for items i_1-i_3.

	Prediction scores for users A-E					**Group recommendation score**	
Item	A	B	C	D	E	Average aggregation	Least misery
$\mathbf{i_1}$	2	4	3	5	1	3	1
$\mathbf{i_2}$	4	1	4	4	5	3.6	1
$\mathbf{i_3}$	3	3	2	3	3	2.8	2

There are obvious drawbacks to both approaches. In case there is an item that the majority of the group prefers, then average aggregation method will give a high score for that item in the group recommendation list disregarding the minority opinion (see item i_2 in Table 1). It is clear that by using average aggregation, user *B* would be quite unsatisfied for item i_2, even though it has a high score in the group recommendation list. On the other hand, the least misery method takes the lowest score given for an item in the group members' individual recommendation lists as the group recommendation score. Thus, one group member can have disproportionate voting weight while also providing low overall preference scores in the group recommendation list (see items i_i and i_2 in Table 1). For

least misery method, item i_2 has a very low score in the group recommendation list, even though it would be very suitable for most users in the group. Stratigi et al. [9] proposed a novel way of dynamically combining the average and least misery aggregation methods. In this thesis, both of the two proposed novel methods utilize average aggregation as the base aggregation method, but with additional adjustments.

Other variations of score aggregation for group recommendations have also been proposed. Ntoutsi et al. [20] proposed to arrange users into clusters of users with similar preferences. Then for each member of the group, collaborative filtering approach is used to get individual recommendations by using the users from the same cluster as the group member. Finally, the individual recommendation lists are aggregated to a group recommendation list with a top-*k* algorithm. In this thesis, both of the two proposed methods use a high similarity threshold value to find the most similar users for a given group member. Then, these most similar users' preferences are used to get individual group member recommendation lists with collaborative filtering. And finally, both of the two proposed methods utilize a novel aggregation method to combine the individual group members' recommendation lists into a group recommendation list. Yuan et al. [7] proposed a model that gives more influence for the users that have more expertise regarding the items that are recommended. The method gives different weights to the members in the aggregation phase according to the member influence. In this thesis, one of the proposed novel methods uses a similar strategy. While Yuan et al. [7] give more weight to those users that have more expertise, the method proposed in this thesis gives more weight to those users that are less-than-average satisfied to the items in the group recommendation list. Kim et al. [6] proposed a two-stage group recommendation method for improving the satisfaction of the group members and recommendation effectiveness. In the first phase, the method uses the virtual user approach to combine group members' profiles and collaborative filtering to generate a group recommendation list for that virtual user. In the second phase, the method removes items from the group recommendation list that are not preferred by the members of the group. In this thesis, one of the proposed methods similarly uses a two-stage process where in the first stage the group recommendation list is generated, and in the second stage, the top items are reordered to minimize disagreement between the individual group members' predicted preference and the score in the group list for each item.

2.3. Multi-round recommendations

In a multi-round recommendation scenario, the recommender system is required to provide recommendations in consecutive rounds. When applying standard recommendation methods to multi-round recommendations, there is an obvious setback: the methods do not take into account the results of the previous recommendation results. There are two main problems when making recommendations for sequence of consecutive rounds. First

of all, it is important to consider recommendations of the previous rounds so as to not recommend the exact same items for the present round. Secondly, there is another problem. If the group members are not equally satisfied for a recommendation result in a given round, there are no guarantees that the degree of satisfaction of the group members would even out during the recommendation sequence, without balancing the aggregation of the group recommendation list. Stratigi et al. [9] showed that standard group recommendation methods are not suitable for multi-round recommendations as they do not consider the degree of satisfaction of each group member during the recommendation sequence.

Quadrana et al. [8] categorized multi-round recommendations to three categories based on the how much information is known on past interactions. The categories are *last-N interactions-based recommendations*, *session-based recommendations* and *session-aware recommendations*. In the first category, only the last N user actions are considered when making the recommendations. The second category is based on the previous sequence of actions only. In the third category, the system has knowledge of both the interactions made in the past and the previous sequence of actions. In this thesis the third approach, session-aware recommendations, is applied when evaluating the performance of three multi-round group recommendation methods.

2.4. Fairness in recommendations

For single-user recommendations a recommender system has a relatively straightforward task: provide the user with recommendations with highest predicted user preference. There is another dimension in group recommendations: how to provide fair recommendation results for all users in the group. Many different approaches can be considered when taking into account the fairness of recommendations for groups. Guzzi et al. [18] proposed an interactive approach in which the group members can communicate to get new recommendations based on the proposals of the other group members. Each member can make proposals regarding other group members' recommendations. Then, each user can get new recommendations similar to these proposals. Machado et al. [22] studied a scenario in which fair recommendations are needed for generating the groups instead of making recommendations for groups. In their scenario there are a set of users with certain skills, and there is a need to assemble teams with multidisciplinary requirements. They argue that for fair overall results it is not wise to put the best users to a given team as those users are then unavailable to other teams. Instead, they propose a way to organize the users into teams in a fair way. For each team, they calculate a score that indicates how well the team members' skills match the team skill requirements. Then, the teams are formed in such a way that minimizes the pairwise difference between the team scores. Kaya et al. [23] proposed a definition of fairness that takes into account the ranking of the items in the group recommendation list. Typically, the result of a group recommendation is an ordered list of most relevant items. According to their definition of fairness,

a top-*N* group recommendation result is fair if each prefix of the top-*N* items is in balance regarding the preferences of each group member. What they mean is that the first item of the list should be balanced between the group members, as well as the first two items of the list when taken together, as well as the first three items of the list when taken together, and so on. Serbos et al. [19] defined two aspects of fairness: *fairness proportionality* and *envy-freeness*. Fairness proportionality defines that a user considers the results fair even though the result contains items with low relevance to that user as long as the result contains at least *m* items with high relevance score to that user. Envy-freeness defines that a user considers the results fair as long as the result contains at least *m* items that the user does not feel envious about. Lin et al. [21] defined *individual utility* as a measure of how relevant the recommended items are to a user and proposed two definitions of *fairness* and *social welfare*. They defined fairness as the degree of imbalance between the group members' individual utilities. Social welfare is the overall utility of the group calculated as the average of the group members' individual utilities for a group recommendation list. Amer-Yahia et al. [5] utilized a *consensus function* that uses a weighted combination of two components, namely relevance and disagreement. In their function, relevance is defined by using standard aggregation methods such as average aggregation. The component for disagreement is defined in their work as average of pairwise differences of the item relevance scores for the group members or the mathematical variance of relevance scores for the item among group members. In this thesis, one of the proposed novel methods, *average-min-disagreement* utilizes similar strategy. While the consensus function uses a weighted combination of relevance and disagreement components, average-min-disagreement method uses a two-stage strategy where the most relevant items are aggregated first, and then the top items are sorted according to lowest disagreement between individual user relevance scores and the group recommendation relevance score. Stratigi et al. [9] proposed a measure of satisfaction for groups in a multi-round group recommendation scenario. They defined that a recommendation result is fair for all members of the group when the result maximizes the overall group satisfaction, that is the average of individual user satisfaction, and minimizes the variance between group members' individual satisfaction scores, defined as group disagreement. This definition of fairness is used in this thesis, when evaluating the performance of two novel multi-round group recommendation methods in comparison with the hybrid aggregation method recently proposed by Stratigi et al. [9].

2.5. Recommendation quality

Recommender systems typically provide a list of recommendations with most relevant items for the user. In order to evaluate the quality of results for a recommender system, [9] Stratigi et al. defined single user satisfaction and group satisfaction as a measure of recommendation result quality. They defined a single-user's satisfaction by comparing

the user's preference to the items in the group recommendation list and the user's preference to the items in the user's individual recommendation list. Furthermore, they defined a group's satisfaction as the average of the single-user satisfaction scores of all the members in the group. In this thesis, the satisfaction measures defined by Stratigi et al. [9] are used to assess the satisfaction of a group and its members to the group recommendation results.

The recommendation lists provided by recommender systems are typically ranked in order of relevance. Ranking quality, that is the order in which the items are presented in the recommendation list, can also be used to measure recommendation result quality. Vilakone et al. [10] and Zhang et al. [17] used Normalized Discounted Cumulative Gain (NDCG) as a method for measuring the ranking quality of a recommender system. The NDCG takes into account the order of the items as they are presented in the recommendation list. It is formally defined as

$$NDCG = \frac{DCG}{IDCG} \tag{1}$$

$$DCG = \sum_{i=1}^{p} \frac{rel_i}{log_2(i+1)} \tag{2}$$

$$IDCG = \sum_{i=1}^{p} \frac{2^{rel_{i-1}}}{log_2(i+1)}, \tag{3}$$

where DCG stands for *discounted cumulative gain* and IDCG is the *ideal discounted cumulative gain*. DCG penalizes more relevant items that do not appear at a high position in the result list by reducing the graded relevance value logarithmically proportional to the position of the item in the recommendation result list. IDCG on the other hand calculates the ideal situation regarding the recommendation list so as if the items in the recommendation list are presented in descending order with regards to the relevance of the items in the list. For a movie recommender the DCG is calculated with the top-*N* items in the actual order as they appear in the recommendation list, while IDCG score is calculated with the top-*N* items of the recommendation list sorted in descending order according to the predicted preference score. Typically, for a single-user recommendation the DCG and IDCG are equal as the item in the recommendation list appear in order of relevance (predicted preference score). For group recommendations, on the other hand, the items in the recommendation list do not appear in order of relevance, from a single user's point of view. Thus, the DCG score can be calculated for each user by using the user's predicted preference scores for the items in the group recommendation list in the order they are presented in the group recommendation list. The IDCG score is then calculated for each user by using the user's predicted preference scores for the items in the group recommendation list in ideal order for that user. In this thesis, the NDCG score is used as an additional recommendation result quality measure in the experimental evaluation of three

group recommendation methods, as it takes into account the order of appearance of the top items in a recommendation result list.

3. Multi-round group recommendations

Three methods are used in this thesis for multi-round group recommendations. Each of the three methods follow a similar overall strategy for making group recommendations. The methods first generate an individual recommendation list for each group member and then utilize score aggregation approach to combine these individual recommendation lists into the group recommendation result. The methods calculate the individual recommendation lists in the same way, but it is the score aggregation part where the methods differ.

Each of the three recommendation methods uses collaborative filtering approach for generating the individual group members' recommendation lists. A collaborative filtering recommender system finds similar users from the system and then recommends such items to the target user for which the similar users have given high ratings. A similar user in this context is a user that has rated identical items with the target user with similar ratings. The degree of similarity of users u and v can be estimated with Pearson correlation [4]

$$sim(u,v) = \frac{\sum_{i \in I}(r_{u,i}-\overline{r_u})(r_{v,i}-\overline{r_v})}{\sqrt{\sum_{i \in I}(r_{u,i}-\overline{r_u})^2}\sqrt{\sum_{i \in I}(r_{v,i}-\overline{r_v})^2}}, \quad (4)$$

where $r_{u,i}$ is the rating of user u for item i, $\overline{r_u}$ is the average rating of user u and the $i \in I$ summations are over the items that both the users u and v have rated. The degree of similarity gets values $sim(u,v) \in [-1,1]$.

In order to generate a recommendation list for the user, a collaborative filtering recommender system predicts preference scores for new items for the user. A weighted aggregation of similar users' ratings can be used as the basis of the preference score prediction. The degree of similarity of other users can be used as the weight in the preference score prediction by utilizing Pearson correlation (Equation 4). The predicted preference score for an item i for user u can be predicted with the Weighted Sum of Others' Ratings [4]

$$pred(u,i) = \overline{r_u} + \frac{\sum_{v \in U} sim(u,v)*(r_{v,i}-\overline{r_v})}{\sum_{v \in U} sim(u,v)}, \quad (5)$$

where $\overline{r_u}$ is the average rating of user u, $sim(u,v)$ is the Pearson correlation similarity value for users u and v, $r_{v,i}$ is the rating of user v for item i and the $v \in U$ summations are over all the users who have rated the item i. The items are then sorted in descending order according to the predicted score for the user. Typically, top-N items are presented to the user from the sorted recommendation list.

In a given recommendation round, each of the three group recommendation methods uses their own way of aggregating the individual group members' recommendation lists

into the group recommendation result. In order to compare the performance of the three proposed recommendation methods, the following measures are used for each method in the experimental study: *group satisfaction*, *group disagreement*, *F-score*, *NDCG score*.

With group satisfaction the idea is to measure how satisfied the group members are to a given group recommendation result as individuals and as a group. Stratigi et al. [9] defined group satisfaction on two levels. For a single user in the group the satisfaction is defined by comparing the quality of the group recommendation result from the user's point of view. The group's satisfaction can then be measured as the average of the single user satisfaction scores. A single user's satisfaction can be estimated by comparing the user's predicted preference scores of the items in the group recommendation result to the user's predicted preference scores of the ideal items for that user. The ideal items are the top items in the user's individual recommendation list. Formal representation for the satisfaction of user u_i for the group recommendation list Gr is [9]

$$sat(u_i, Gr) = \frac{GroupListSat(u_i, Gr)}{UserListSat(u_i, A_{u_i})} \quad (6)$$

$$GroupListSat(u_i, Gr) = \sum_{i \in Gr} p(u_i, i) \quad (7)$$

$$UserListSat(u_i, A_{u_i}) = \sum_{i \in A_{u_i}} p(u_i, i), \quad (8)$$

where Equation (7) is the sum of predicted preference scores for user u_i for the top-N items in the group recommendation list Gr and Equation (8) is the sum of predictions scores for the top-N items in the individual recommendation list A_{u_i} for user u_i. Now, with single user satisfaction from Equation 6, the satisfaction of group G for a group recommendation list Gr is [9]

$$groupSat(G, Gr) = \frac{\sum_{u_i \in G} sat(u_i, Gr)}{|G|}, \quad (9)$$

that is the average satisfaction of group G.

For a multi-round recommendation scenario, it is important to also consider the group satisfaction over a sequence of recommendations. The overall satisfaction of a group G after k recommendation rounds can be defined as

$$groupSatO(G, k) = \frac{\sum_{j=1}^{k} groupSat(G, Gr_j)}{k}, \quad (10)$$

where Gr_j is the group recommendation list at recommendation round j.

With group disagreement the idea is to measure how satisfied the group members are compared to other group members for a given group recommendation list. A low group disagreement score indicates that the group members are evenly satisfied whereas a high

group disagreement indicates the opposite. In the experimental study in this thesis, the group disagreement definition defined by Stratigi et al. [9] is used to measure group disagreement. Group disagreement can be estimated as the difference of the satisfaction scores of the most satisfied and least satisfied users in the group. The disagreement of group *G* for group recommendation list *Gr* is defined as [9]

$$groupDis(G, Gr) = max_{u_i \in G} sat(u_i, Gr) - min_{u_i \in G} sat(u_i, Gr), \quad (11)$$

where $sat(u_i, Gr)$ is the satisfaction of user u_i that is defined in Equation 6. High disagreement score indicates that there is a large difference between the satisfaction scores of the most satisfied user and the least satisfied user in the group. A low disagreement score, on the other hand, indicates that there is a small difference in these satisfaction scores, and by definition, a small satisfaction score difference between all group members.

For a multi-round recommendation scenario, it is important to also consider the group disagreement over a sequence of recommendations. The overall disagreement of a group *G* after *k* recommendation rounds can be defined as

$$groupDisO(G, k) = \frac{\sum_{j=1}^{k} groupDis(G, Gr_j)}{k}, \quad (12)$$

where Gr_j is the group recommendation list at recommendation round *j*.

As method *A* might produce higher satisfaction scores than method *B* while method *B* could produce lower group disagreement score than method *A*, another measure is needed to be able to compare such two methods. Stratigi et al. [9] proposed *F-score* as a measure for combining group satisfaction and disagreement scores. F-score is formally defined as [9]

$$F\text{-}score = 2\frac{groupSatO * (1 - groupDis)}{groupSatO + (1 - groupDis)}. \quad (13)$$

As each of the three methods provide a group recommendation list where the top-N items are ordered according to the relevance score, an additional measure is also used in the experimental study in this thesis for comparing the ranking quality of the methods from a single user's point of view. For measuring the ranking quality of a group recommendation list during the recommendation sequence, the NDCG score introduced in Equation 1 is calculated after each recommendation round for each group member. Then, to be able to consider the NDCG from the group's point of view, the group NDCG score for group *G* and group recommendation list *Gr* is defined with Equation 1 as

$$groupNDCG(G, Gr) = \frac{\sum_{u_i \in G} NDCG(u_i, Gr)}{|G|}, \quad (14)$$

that is the average NDCG score for group *G*.

For a multi-round recommendation scenario, it is important to also consider the group NDCG score over a sequence of recommendations. The overall NDCG score of a group *G* after *k* recommendation rounds can be defined as

$$groupNDCGO(G, k) = \frac{\sum_{j=1}^{k} groupNDCG(G, Gr_j)}{k}, \quad (15)$$

where Gr_j is the group recommendation list at recommendation round *j*.

4. Methods

In this chapter we introduce and present the details of all three methods proposed for realizing multi-round group recommendations. In Section 4.1. we introduce the sequential hybrid aggregation method proposed by Stratigi et al. [9]. In Sections 4.2.-4.3., we introduce two novel group recommendation methods: adjusted average aggregation method and average-min-disagreement aggregation method.

4.1. Sequential hybrid aggregation

The sequential hybrid aggregation (SHA) method is a multi-round group recommendation method that uses a weighted combination of average aggregation and least misery aggregation approaches and dynamically adjusts the weights during a group recommendation sequence. The SHA method aims to provide highly relevant results to the group while being fair for all users of the group. The SHA method uses a modified least misery aggregation, where the least misery function returns the least satisfied user's predicted score for a given item. By giving more influence to the least satisfied user, the SHA method aims to keep all group members satisfied. SHA calculates the preference score for group *G* for item *i* at iteration *j* and is formally defined as

$$score(G,i,j) = \left(1-\alpha_j\right) * avgScore(G,i,j) + \alpha_j * leastScore(G,i,j), \quad (16)$$

where $avgScore(G,i,j)$ is the score of item *i* as it is computed by average aggregation method during iteration *j*, and $leastScore(G,i,j)$ is the least satisfied user's score of item *i* at iteration *j*. The weight α is calculated dynamically at each iteration by subtracting the minimum satisfaction score of the group members in the previous iteration, from the maximum satisfaction score as

$$\alpha_j = max_{u\in G} sat\left(u, Gr_{j-1}\right) - min_{u\in G} sat(u, Gr_{j-1}). \quad (17)$$

The SHA method was shown to outperform typical group recommendation methods in a multi-round group recommendation scenario [9]. Nevertheless, there remain some possible improvements from the SHA method. First of all, the SHA method only considers one unsatisfied user (the least satisfied user). In a scenario where there are multiple unsatisfied users, other approaches might perform better when trying to get high quality results for the group while ensuring fairness for the whole group. In Section 4.2. we propose a novel multi-round group recommendation method that aims to achieve that by giving more influence to all users that are less satisfied that on average in the group. Secondly, in Section 4.3 we propose another novel method that aims to outperform the SHA method with a slightly different approach: by recommending to the group items that

are highly relevant but for which there is small difference in the aggregated group score and the individual group members' predicted preference scores.

4.2. Sequential adjusted average aggregation

Sequential adjusted average aggregation (SADJA) method is one of the two novel multi-round group recommendation methods. While SHA only considers one unsatisfied user in the group, SADJA is designed to consider multiple unsatisfied users in a group. The main idea in SADJA is that it gives more influence to those group members, whose satisfaction score in the previous recommendation round was less than the average satisfaction score in that round. While SHA picks the least satisfied user's predicted score and then aggregates the group score as a combination of the least satisfied user's score and standard average aggregation score, SADJA, on the other hand, considers all such users unsatisfied, who are less satisfied than the group average. Then, SADJA utilizes standard average aggregation to calculate group recommendation scores for items in the group recommendation list but gives extra weight to all unsatisfied users.

The SADJA method is based on weighted average aggregation in which the weights are adjusted dynamically according to group member satisfaction scores. The SADJA method calculates for group *G* the preference score for item *i* at iteration *j* and can be formally defined as

$$scoreSADJA(G,i,j) = \frac{\sum_{u_i \in G} w_{u_i,j} * pred(u,i)}{|G|}, \tag{18}$$

where $pred(u,i)$ is the predicted preference score for user *u* for item *i* (Equation 5) and $w_{u_i,j}$ is the weight for user *u* at iteration *j* according to the satisfaction scores from previous round. The weight *w* is defined as

$$w_{u_i,j} = \begin{cases} 1, & if\ sat(u_i, Gr_{j-1}) > groupSat(G, Gr_{j-1}) \\ 1 + \omega_{u_i,j}, & otherwise \end{cases}, \tag{19}$$

where $sat(u_i, Gr_{j-1})$ is the satisfaction score for user *u* for group recommendation list *Gr* at iteration *j-1* (Equation 6), $groupSat(G, Gr_{j-1})$ is the average group satisfaction score for group *G* for group recommendation list *Gr* at iteration *j-1* (Equation 9) and $\omega_{u_i,j}$ is a calculation coefficient for user *u* at iteration *j* that is defined as

$$\omega_{u_i,j} = \begin{cases} \omega_0 * f_{u_{j-1}}, & if\ pred(u_i,i) > avgScore(G,i,j) \\ -\omega_0 * f_{u_{j-1}}, & otherwise \end{cases}, \tag{20}$$

where ω_0 is a static coefficient and $f_{u_{j-1}}$ is defined as the difference between the group's average satisfaction and the satisfaction score of the user *u* at iteration *j-1* using Equations 6 and 9

$$f_{u_{j-1}} = groupSat\left(G, Gr_{j-1}\right) - sat\left(u_i, Gr_{j-1}\right). \quad (21)$$

From Equation 19 it can be seen that for a user with a higher satisfaction than the average from the previous iteration, the weight $w_{u_i,j} = 1$ and that user will get no additional influence when aggregating the group recommendation list with Equation 18. But for those users that have a lower satisfaction score than average from the previous round, they will get more additional influence in Equation 18 determined by $\omega_{u_i,j}$. The coefficient ω_0 is a static value: various values for ω_0 were experimented before the actual group recommendation method comparison with $\omega_0 \in$ [0, 0.25, 0.5, 1.25, 1.5, 2]. These pre-experiments indicated that coefficient ω_0 = 0.25 would give the best overall results for SADJA method. In Equation 20, the calculation coefficient $\omega_{u_i,j}$ is positive if the user's predicted preference for item *i* is higher that the group's average predicted preference for that item (the additional influence for that user in Equation 18 becomes $w_{u_i,j} = 1 + \omega_0 * f_{u_{j-1}}$, and the coefficient is negative if the user's predicted preference is for item *i* is lower that the group's average predicted preference for that item (the additional influence for that user in Equation 18 becomes $w_{u_i,j} = 1 - \omega_0 * f_{u_{j-1}}$). As the $\omega_{u_i,j}$ coefficient is only considered for users with lower satisfaction than average (Equation 19), the idea is to give more influence to the less than average satisfied users by "pushing" the average rating of item *i* more towards their individual predicted preference of that item.

With Equation 21 it can be seen that the degree of additional influence given to the less than average satisfied group members depend on the magnitude of the difference between the satisfaction score of that user's satisfaction score and the group's average satisfaction. So, a user with a very low satisfaction score compared to the average satisfaction in the previous round will get more additional influence in the current round (higher value for $f_{u_{j-1}}$) compared to a user whose satisfaction is only slightly lower than the average satisfaction in the previous round.

4.3. Average-min-disagreement aggregation

Average-min-disagreement aggregation (AMD) method is the second of the two novel group recommendation methods. The AMD method has a different approach for the score aggregation phase of the group recommendation procedure compared to the SHA and SADJA methods. Whereas SHA and SADJA concentrate on giving more influence for the less satisfied users in a group, the AMD method aims to minimize the difference between an item's group score and the individual group members' individual scores for that item and through that provide acceptable results for the whole group. In contrast to SHA

and SADJA methods, the AMD method treats all group members similarly in the group aggregation phase whether or not they were satisfied in the previous round.

The AMD method is a two-stage process. It is not a multi-round group recommendation method as such, but instead it can be used for both single-use recommendation scenarios and multi-round recommendation scenarios. In the first stage, the group recommendation list is generated using standard average aggregation method. In the second stage, the AMD method takes the top-K items from the group recommendation list and reorders them by minimizing the difference between the group score and the group members' predicted scores for the items. The AMD method's main emphasis is to provide recommendations with low disagreement between the group members regarding the items in the final group recommendation list. And by taking the top-K items generated by average aggregation in the first stage before reordering them, AMD is still able to provide highly relevant items in the group recommendation result list.

Various values for K were experimented before the actual group recommendation method comparison with $K \in [50, 100, 200, 300]$. These pre-experiments indicated that, in the scenario used in the experiment, choosing a high K value would lead to more relevant results in the group recommendation list (higher group satisfaction) while a lower K value would lead to lower disagreement between the group members regarding the group recommendation list (lower group disagreement). These pre-experiments indicated that $K = 200$ would give the best overall results for AMD method.

So, after the first stage of AMD method, the group recommendation list consists of the top-K items according to the predicted group preference score (aggregated as average of group members' individual predicted preferences). At the second stage the method calculates *score disagreement* for each of the top-K items. For each of these items, the method calculates the difference between the group score and each group members' individual predicted score, that is, the score disagreement between each users' individual predicted preference score and the predicted group score (Equation 22). With these score disagreement values, the method then calculates for each item the group score disagreement value (Equation 25) by subtracting the item's minimum score disagreement value (Equation 24) from the maximum score disagreement value (Equation 23). Finally, the AMD method then reorders the top-K items in descending order according to the group score disagreement value. The final group recommendation list then consists of highly relevant items with low disagreement between the group members' individual predicted scores.

The steps of the second stage of the AMD method can be with Algorithm 1 defined as

Algorithm 1. The steps of the second stage of the AMD method.

Step 1: $predScoreDis(u,i) = \left|\frac{pred(u,i)}{avgScore(G,i)} - 1\right|$ (22)

Step 2.1: $maxScoreDis(G,i) = max_{u \in G} predScoreDis(u,i)$ (23)

Step 2.2: $minScoreDis(G,i) = min_{u \in G} predScoreDis(u,i)$ (24)

Step 3: $groupScoreDis = maxScoreDis(G,i) - minScoreDis(G,i)$, (25)

where Step 1 calculates the disagreement for user *u*'s predicted score for item *i*, where $pred(u,i)$ is the user's predicted preference score for the item (Equation 5) and $avgScore(G,i)$ is for group *G* the average aggregation score for item *i*. Steps 2.1 and 2.2 calculate for group *G* the maximum and minimum score disagreement for item *i* using Equation 22. Step 3 then calculates for group *G* the group score disagreement for item *i* by subtracting the maximum score disagreement from the minimum score disagreement. From Step 3 it can be seen that it follows similar strategy that is used also for group disagreement (satisfaction-wise) calculation with Equation 11. After executing steps 1-3 for each item in the top-*K* group recommendation list, AMD method reorders the *K* items in order of ascending $groupScoreDis$ values.

5. Experimental setup

Two novel multi-round group recommendation methods, *sequential adjusted average aggregation* (SADJA) and *average-min-disagreement aggregation* (AMD) were proposed in Sections 4.2. and 4.3. An experiment was designed to study the performance of these two novel methods. In addition to the two proposed methods, the recently introduced *sequential hybrid aggregation* (SHA) method (see Section 4.1) was used in the experiment as a baseline method for multi-round group recommendation performance.

For each of the above-mentioned recommendation methods, the recommendation procedure is the same. Each recommendation method is applied to a stable group of five users for a 15-round group recommendation sequence. After each round, the recommendation method recommends the group the 10 items with the highest group preference score. The items recommended by a method in a given round will not be recommended again by that method in the following rounds, so as not to recommend same movies more than once. In practice, the three recommendation methods (SADJA, AMD, SHA) were computed in parallel for each group, but they were handled separately with no effects between the methods.

In order to evaluate the methods in different scenarios, distinct types of groups were used (see Section 5.1) with different degrees of similarity between the users in a group. For each group type, 100 groups were generated. The results for each group type were calculated as an average over 100 groups.

A Python script was developed by the author to handle the experiment procedure. The experiment was executed in two stages. In the first stage, the script was used to generate 100 groups for each of the selected group types (see Section 5.1). Algorithm 2 illustrates the first stage of the experiment.

Algorithm 2. Pseudocode for the first stage of the evaluation procedure: generating the groups for each group type (see Section 5.1.).

```
GROUPTYPES = [all-similar, 4+1, 3+2, 3+1+1, all-dissimilar]                  // a list
GROUPS = {all-similar : [], 4+1 : [], 3+2 : [], 3+1+1 : [], all-dissimilar : []} // a dict
for type in GROUPTYPES:
    grouplist = []                      // initialize an empty list
    for i = 1...100:                    // repeat 100 times (create 100 groups)
        group = createGroup()           // generate group (where group type = type)
        grouplist.append(group)         //  add generated group to the list
    GROUPS[type] = grouplist            // finally, add generated list of groups to the dict
```

In the second stage, the script was used to perform group recommendations for the generated groups one by one. For each group, the group recommendations were performed in a sequence consisting of 15 rounds. At each round, group recommendations were calculated with each method. Then, after each round, the script calculates group satisfaction, group disagreement, F-score and NDCG score for each method. Algorithm 3 illustrates the second stage of the experiment procedure.

Algorithm 3. Pseudocode for the second stage of the evaluation procedure: calculating group recommendations for each group (see Algorigthm 2) with multi-round group recommendation methods (see Section 4.1.-4.3).

```
for type in GROUPS.Keys():          // go through group types one by one
   groups = GROUPS[type]            // get the groups (a list of 100 groups)
   for group in groups:             // loop through the groups one by one
      for i = 1…15:                 // go through 15 recommendation rounds
         for method in METHODS:     // use the methods one by one
            calculate and save      // calculate and save results for each method
```

The data set used in the experiment and additional constraints are explained in Sections 5.2.-5.3. The results of the experiment are presented in Section 6 with analysis and comparison of method performance.

5.1. Group types

In the experiment the group recommendations are calculated for stable groups consisting of five users. Stratigi et al. [9] used several types of small groups (groups of five users) with different degrees of similarity between the group members in an experimental study of different multi-round recommendation methods. Each of those group types included at least one user that was dissimilar with the other users in a group. In this thesis, the same types of groups are implemented in the experiment as those can be used to evaluate group recommendation methods in diverse scenarios. In addition to the group types used by Stratigi et al. [9], one additional group type was used in the experiment in this thesis: a group type in which all users in the group are similar with each other.

Formally, the experiment was performed utilizing distinct group types where each group is formed from subgroups with the following characteristics:

- Each user in a subgroup shares a similarity value higher than $c_{group\text{-}similarity}$.
- Users from different subgroups share a similarity value lower than $c_{group\text{-}disimilarity}$.

Five distinct group types fulfilling these characteristics were used: *all-similar, 4+1, 3+2, 3+1+1*, *all-dissimilar*. The group type names represent the composition of each group

type. The group types are explained below in table Table 2. 100 groups were generated from the ratings data for each group type.

Table 2. The groups used in the experiment.

Group type	Subgroup analogy	Explanation
all-similar	A group consisting of one subgroup with 5 users.	Each user is similar with each other.
4+1	A group consisting of two subgroups with 4 and 1 users.	4 users are similar with each other, 1 user is dissimilar with all users.
3+2	A group consisting of two subgroups with 3 and 2 users.	3 users are similar with each other (subgroup 1), 2 users are similar with each other (subgroup 2). Users in subgroup 1 are dissimilar with users in subgroup 2.
3+1+1	A group consisting of three subgroups with 3, 1 and 1 users.	3 users are similar with each other. The other 1+1 users are dissimilar with all users.
all-dissimilar	A group consisting of 5 subgroups with 1 user in each subgroup.	Each user is dissimilar with each other.

A sample similarity matrix for a generated group is shown in Table 3. The table indicates that users A, B and E are similar with each other with similarity values higher than $c_{group\text{-}similarity}$. User C is dissimilar with all users with similarity values lower than $c_{group\text{-}dissimilarity}$. User D is also dissimilar with all users with similarity values lower than $c_{group\text{-}dissimilarity}$.

Table 3. A sample similarity matrix for a group *3+1+1* with users *A, B, C, D, E* and similarity values between users with $c_{group\text{-}similarity} = 0.5$ and $c_{group\text{-}dissimilarity} = -0.5$.

	A	**B**	**C**	**D**	**E**
A		0.631	-0.655	-0.716	0.763
B	0.631		-0.558	-0.671	0.746
C	-0.655	-0.558		-0.602	-0.502
D	-0.716	-0.671	-0.602		-0.719
E	0.763	0.746	-0.502	-0.719	

5.2. Additional constraints

Two additional constraints were used when considering the similarity of users from the ratings data. As there are over 70 000 users in the chosen data set, it is not necessarily practical to calculate similarity value for each user in the data set. Secondly, in case there

are two users u_1 and u_2, who only have rated very few identical movies, the Pearson correlation value for these two users do not necessarily give a very accurate estimate on the true similarity of these two users. Therefore, in the scope of this work, a constraint m for the minimum number of identical movies rated by two users was implemented when considering the similarity of two users. When considering user u and users U, the similarity value is only calculated for such users from U that have rated at least $m = 6$ identical movies with user u, where users U are all the users in the ratings data except user u. The similarity value is calculated with Equation 4.

In addition, a correlation threshold c was used to prune similar and dissimilar users from users U who already fulfill the m constraint. In the scope of this work, similar users are the users who share a similarity value higher than $c_{similarity}$. Dissimilar users are the users who share a similarity value lower than $c_{dissimilarity}$. Different values for the correlation threshold c were implemented in the experiment:

- Two users in a group are considered similar when they share a similarity value higher than $c_{group\text{-}similarity} = 0.5$.
- Two users in a group are considered dissimilar when they share a similarity value lower than $c_{group\text{-}dissimilarity} = -0.5$.
- When calculating recommendations for a single user with Equation 5, a higher similarity value threshold was used to get higher quality predictions. In this context users from U are considered similar with the target user u when they share a similarity value higher than $c_{similarity} = 0.7$.

For SADJA method there is also the adjustable static coefficient ω_0. For the experimental study $\omega_0 = 0.25$ was used (see Section 4.2.). For AMD method there is also one adjustable parameter K that is used to take the top-K items from the group recommendation result calculated in the first stage of method AMD before the second stage. For the experimental study $K = 200$ was used (see Section 4.3.). The main constraints are summarized in Table 4.

Table 4. Additional constraints used in the experiment.

Constrain	Value	Explanation
m	6	The minimum number of identical movies rated by two users.
$c_{group\text{-}similarity}$	0.5	The similarity threshold for similar users in a group.
$c_{group\text{-}dissimilarity}$	-0.5	The dissimilarity threshold for dissimilar users in a group.
$c_{similarity}$	0.7	The similarity threshold for predicting recommendations.
ω_0	0.25	Static coefficient used for SADJA method.
K	200	The parameter used for AMD (top-K items).

5.3. Data set

GroupLens from the Department of Computer Science and Engineering at the University of Minnesota provides several data sets of different sizes for movie ratings. This group provides the MovieLens movie rating data sets with 100 000, 1 million, 10 million and 20 million ratings, among others. The data set with 10 million ratings was decided to be sufficiently large for the experiment. The data used in the experiment was the MovieLens 10M Dataset [1] with 10 000 054 ratings of 10 681 movies. The data set contains ratings by 71 567 users where each user has rated at least 20 movies. The ratings are made on a scale of 0.5 - 5.0 stars with half-star increments.

6. Results

To be able to compare the performance of the three group recommendation methods (SHA, SADJA, AMD), below is presented the results for group satisfaction (Equation 10), group disagreement (Equation 12), F-score (Equation 13) and group NDCG score (Equation 15) for each method. Each of these measures are calculated after 5, 10 and 15 recommendation rounds and the results are aggregated as an average of 100 groups. In addition, the results of the experiment are presented separately for each group type. For example, Figure 1.1 shows group satisfaction results for group type *all-similar* and it can be seen that with both SHA and SADJA methods the group satisfaction after 5 recommendation rounds is approximately 0.8 on average over 100 groups.

Group satisfaction

Figures 1.1-1.5 show the results for group satisfaction for all five group types. Group satisfaction follows a similar pattern in all group type scenarios for all three methods: for AMD the group satisfaction remains approximately constant through the 15-round recommendation sequence while for SHA and SADJA methods the group satisfaction decreases almost linearly in each scenario. Methods SHA and SADJA provide higher group satisfaction than AMD across all group types, according to Figures 1.1-1.5. Group satisfaction with methods SHA and SADJA seem to follow almost an identical pattern.

Overall, it seems that the level of group satisfaction is the highest for group type *all-similar* and the lowest for group type *all-dissimilar* while the group satisfaction results are approximately the same with group types *3+2* and *3+1+1* with all three methods.

Group disagreement

Figures 2.1-2.5 show the results for group disagreement for all five group types. Again, group disagreement results follow a similar pattern across all group type scenarios for all three methods: for SHA and SADJA method the group disagreement remains relatively constant trough the 15-round recommendation sequence while the group disagreement slightly increases as the recommendation sequence progresses.

AMD provides the lowest group disagreement, according to Figures 2.1.-2.5., while SHA provides the highest group disagreement in all group type scenarios and at all recommendation sequence intervals. SADJA has slightly lower group disagreement values than SHA, with approximately a 0.10-point difference in each data point in the results.

Overall, it seems that group disagreement is the highest for group type *all-dissimilar* and the lowest for group type *all-similar* with all methods while the group disagreement results are approximately the same with group types *3+2* and *3+1+1* with all three methods.

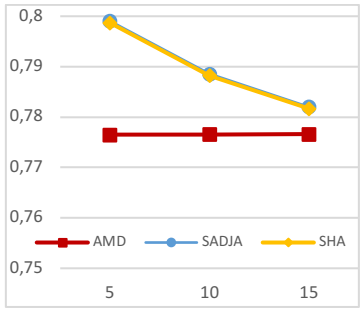

Figure 1.1. Group satisfaction: group type *all-similar* (100 groups average).

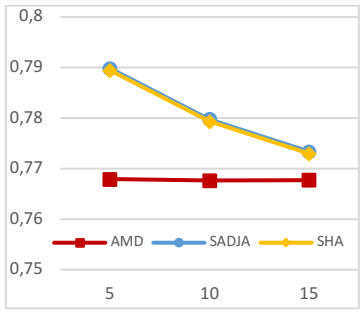

Figure 1.2. Group satisfaction: group type 4+1 (100 groups average).

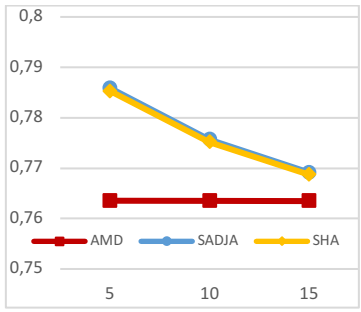

Figure 1.3. Group satisfaction: group type *3+2* (100 groups average).

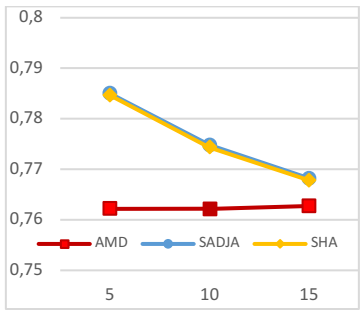

Figure 1.4. Group satisfaction: group type *3+1+1* (100 groups average).

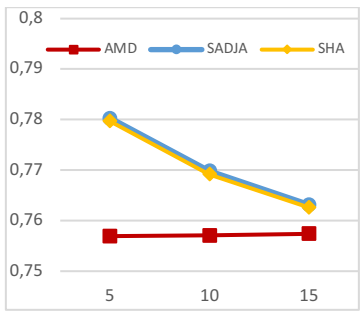

Figure 1.5. Group satisfaction: group type *all-dissimilar* (100 groups average).

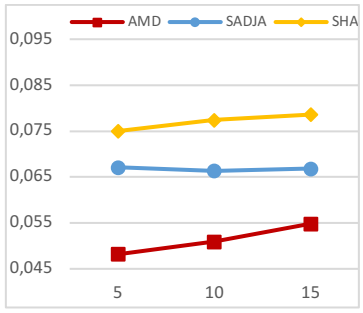

Figure 2.1. Group disagreement: group type *all-similar* (100 groups average).

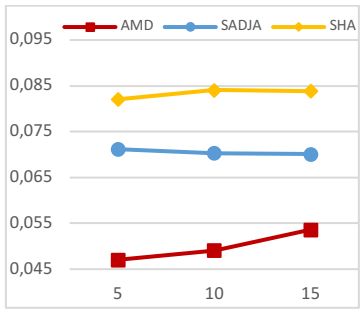

Figure 2.2. Group disagreement: group type *4+1* (100 groups average).

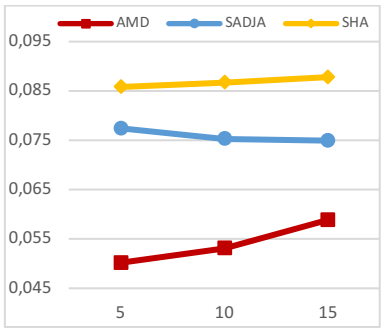

Figure 2.3. Group disagreement: group type *3+2* (100 groups average).

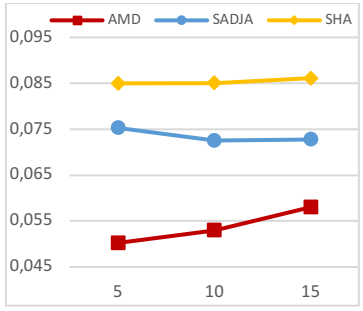

Figure 2.4. Group disagreement: group type *3+1+1* (100 groups average).

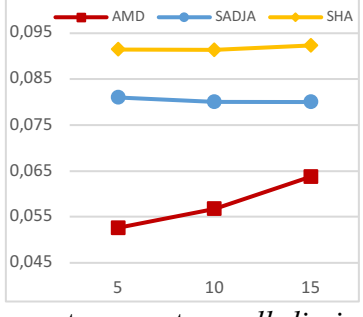

Figure 2.5. Group disagreement: group type *all-dissimilar* (100 groups average).

F-score

Figures 3.1-3.5 show the results for F-score for all five group types. For all three methods the F-score decreases as the recommendation sequence progresses: for AMD the F-score remains almost constant with only a slight decrease during the 15-round recommendation sequence while for SHA and SADJA there is a steeper reduction in the F-score values.

Figures 3.1-3.5 show that after a 5-round recommendation sequence, SADJA achieves the highest F-score values with each group type. In all group type scenarios, SHA has the lowest F-score except for group type *all-similar* (Figure 3.1.) where the AMD has the lowest F-score value.

In three of the five group types (*4+1*, *3+2*, *all-dissimilar*), AMD has the highest F-score after a 10-round recommendation sequence, while AMD and SADJA achieve approximately the same F-score with group types *all-similar* and *3+1+1*. As the recommendation rounds progress, the F-score value for SADJA decreases more steeply than for AMD and after a 15-round recommendation sequence, AMD achieves the highest F-score values with all group types. The SHA method provides the lowest F-score values at all data points, except for after a 5-round recommendation sequence with group type *all-similar*, where it has the second highest F-score, according to Figure 3.1.

Overall, it seems that F-score is the highest for group type *all-similar* and the lowest for group type *all-dissimilar* while the F-score results for group types *3+2* and *3+1+1* are approximately the same.

Normalized Discounted Cumulative Gain (NDCG)

Figures 4.1-4.5 show the results for NDCG score in all group type scenarios. For AMD the NDCG score decreases as the recommendation rounds progress, while on the contrary, the NDCG score increases for methods SHA and SADJA. According to Figures 4.1.-4.5., AMD achieves the highest NDCG score values in all scenarios, even though the NDCG score curve has a relatively steep decreasing curve. The NDCG score values for SHA and SADJA are almost the same, with SHA method providing slightly higher NDCG score in all data points.

Overall, it seems that the NDCG score is the highest for group type *all-similar* and the lowest for group type *all-dissimilar* for all methods. NDCG score results for group types *3+2* and *3+1+1* are approximately the same.

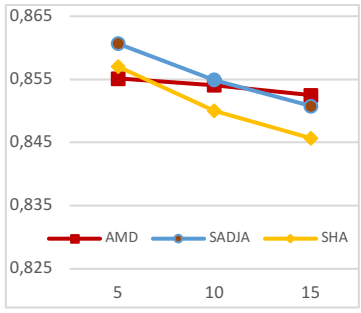

Figure 3.1. F-score: group type *all-similar* (100 groups average).

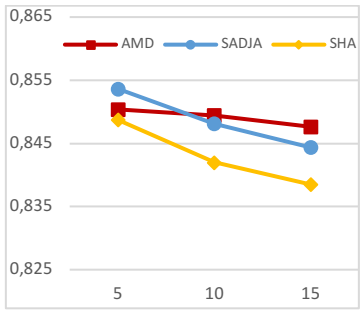

Figure 3.2. F-score: group type *4+1* (100 groups average).

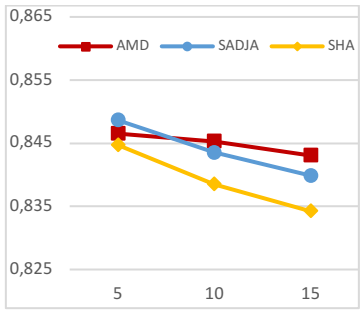

Figure 3.3. F-score: group type *3+2* (100 groups average).

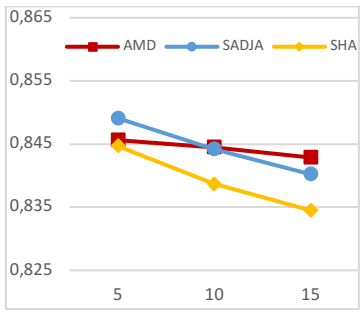

Figure 3.4. F-score: group type *3+1+1* (100 groups average).

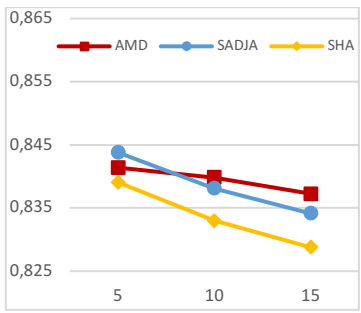

Figure 3.5. F-score: group type *all-dissimilar* (100 groups average).

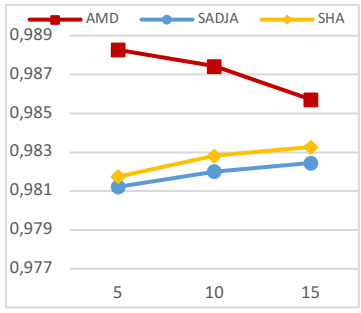

Figure 4.1. NDCG score: group type *all-similar* (100 groups average).

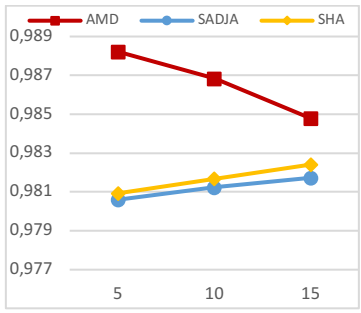

Figure 4.2. NDCG score: group type *4+1* (100 groups average).

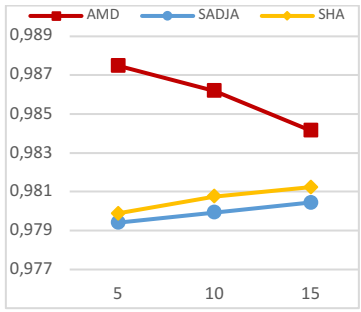

Figure 4.3. NDCG score: group type *3+2* (100 groups average).

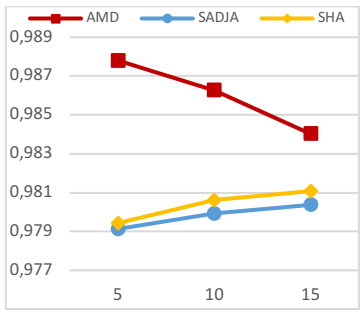

Figure 4.4. NDCG score: group type *3+1+1* (100 groups average).

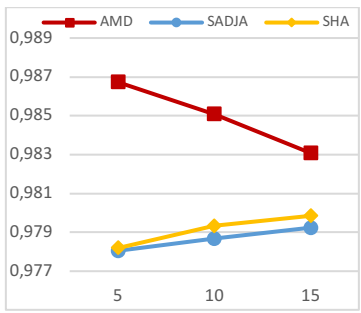

Figure 4.5. NDCG score: group type *all-dissimilar* (100 groups average).

6.1. Analysis

Both of the proposed two novel group recommendation methods, SADJA and AMD, were presented in Section 4. with possible improvements in performance in comparison with the SHA method, with regards to group satisfaction and group disagreement. Extensive experiments were carried out to compare the performance of the three methods in varied scenarios with five distinctly different group types.

Some interesting observations can be made from the results of the experiment. First of all, while SHA and SADJA achieve high group satisfaction results, the group satisfaction decreases when the recommendation rounds progress, according to Figures 1.1.-1.5. AMD on the other hand does not achieve as high group satisfaction results as the other two methods, but it can provide uniform group satisfaction throughout the 15-round recommendation sequence in all group type scenarios.

Secondly, for SADJA and SHA methods the group disagreement remains relatively constant through the 15-round recommendation sequence. This kind of behavior might be due to the fact that both SHA and SADJA method utilize dynamic adjustment of the weights in score aggregation to balance the disagreement in the group during the recommendation sequence. On the other hand, with AMD the group disagreement increases as the recommendation sequence progresses. AMD method does not consider individual group members' satisfaction or disagreement as such but concentrates on the score disagreement, which might be the reason to increasing group disagreement as the recommendation rounds progress. Nevertheless, it is important to note that regardless of the increasing group disagreement, the AMD method still achieves the lowest group disagreement values in all group types. It is particularly notable that AMD achieves the lowest group disagreement values, even though it is "blind" to group member satisfaction and disagreement from previous recommendation rounds.

In addition, it is interesting to note that with all three methods and with all four measures (group satisfaction, group disagreement, F-score, NDCG score) the result patterns do not vary much between the different group types. The levels of the result scores differ as expected (higher group satisfaction for group type *all-similar* than for group type *all-dissimilar*, and the opposite for group disagreement, for example), but each method perform in a similar manner regardless of the group type.

According to group satisfaction results (Figures 1.1.-1.5.) and group disagreement results (Figures 2.1.-2.5.) the results for SHA and SADJA follow a similar pattern in all scenarios. This is likely due to the similar score aggregation strategy utilized by both methods. As explained in Section 4., both SADJA and SHA utilize score aggregation with dynamically changing weights and give more influence to one unsatisfied user (SHA) or all unsatisfied users (SADJA). The approach used by SADJA seem to work better: the group satisfaction is equally high with SADJA and SHA methods, according to group

satisfaction results in Figures 1.1.-1.5., while the group disagreement results are lower for SADJA in all group type scenarios and at each step of the 15-round recommendation sequence. As SADJA achieves lower group disagreement with equally high group satisfaction compared to SHA, it is not surprising that the F-score is also higher for SADJA method in all scenarios.

The comparison between SHA and AMD is not as straightforward. According to group satisfaction results (Figures 1.1.-1.5.), SHA outperforms AMD in all scenarios. On the other hand, AMD achieves clearly lower group disagreement in all scenarios, according to Figures 2.1.-2.5. The F-score is useful as it can be used to consider both group satisfaction and group disagreement at the same time: Figures 3.1.-3.1. show that AMD achieves higher F-score in all group types, and at each step of the 15-round recommendation sequence. These results suggest that AMD has better overall performance with very low group disagreement but with a small sacrifice in group satisfaction. It seems that the strategy of AMD works as designed and explained in Section 4.: to provide highly relevant recommendations with low group disagreement.

The comparison of SADJA and AMD is similar to the comparison of SHA and AMD: SADJA achieves higher group satisfaction scores while AMD achieves lower group disagreement. When comparing the performance of SADJA and AMD with F-score, the result is not obvious. Figures 3.1.-3.5. indicate that for small number of recommendation rounds, SADJA method would be a safe bet as it outperforms AMD in all group types. On the other hand, it seems that for longer multi-round recommendation sequences, AMD could be a better choice as it performs better than SADJA in the later recommendation rounds.

The results for NDCG scores in Figures 4.1.-4.5. show that AMD method achieves clearly higher NDCG score values in all scenarios, compared to SHA and SADJA. This does not come as a surprise, as the NDCG is a measure of recommendation result quality with regards to the order of the items in the result list. As the AMD method sorts the top-K items in the result list according to score disagreement, it seems that it manages to provide results where the order of the items is more favorable to the group, compared to SHA and SADJA methods.

One interesting observation is that for each method, the method performs in a similar pattern across all group types regardless of if the group members are similar or dissimilar. This indicates that the results of this experiment should give valuable information when choosing a multi-round group recommendation method, even if the degree of similarity between a group's users is not known.

Overall, the results show that the two proposed novel group recommendation methods outperform the recently introduced SHA method. In our experiment, both of the proposed methods achieve higher F-score results with five different group types for a 15-round

recommendation sequence. In addition, the results suggest that in some cases the choice of a recommendation method for a multi-round group recommendation sequence depends on what is the main emphasis of the recommendation procedure. For highest group satisfaction with good overall results, the proposed SADJA method is the best choice. But if low group disagreement is wanted while maintaining high group satisfaction, the proposed AMD method is the best choice.

7. Conclusions

In this thesis, two novel group recommendation methods were proposed: sequential adjusted average aggregation (SADJA) and average-min-disagreement aggregation (AMD). SADJA aims to give more influence to those users in the group that are less satisfied than average by utilizing weighed average aggregation, where the weights are adjusted dynamically during the recommendation sequence. AMD introduced a measure of *score disagreement*, that is the difference between the individual group members' predicted preference score for an item and the predicted group score for that item. AMD utilizes a two-stage procedure where highly relevant items for the group are searched first and the top items are then reordered by minimizing the score disagreement.

An experiment was designed to evaluate the performance of these two methods. Several types of groups with different degrees of similarity between the group members were used in the experiment to test the methods in various scenarios for a 15-round recommendation sequence. A recently introduced sequential hybrid aggregation (SHA) method was used in the experiment as the baseline method for current multi-round group recommendations. The performance of the methods was measured using group satisfaction, group disagreement, F-score and Normalized Discounted Cumulative Gain (NDCG). NDCG measure was implemented to test the effect of the order of the items in the group recommendation list at a given round, as the other methods only consider the group recommendation list as unorganized, so as they do not give any importance to the ordering of the items in the result list.

The results of the experiment show that SADJA achieves its target, that is, low group disagreement, while maintaining highly relevant results for the group. The results also show that AMD achieves very low group disagreement across all group types with only a small sacrifice in group disagreement. In addition, the experimental results show that both SADJA and AMD methods outperform SHA across all group type scenarios. According to the results, SADJA method provides better overall results than AMD in the earlier recommendation rounds, but AMD outperforms SADJA in the latter rounds. On the other hand, if minimizing group disagreement is the main emphasis when choosing a group recommendation method for a multi-round recommendation scenario, then AMD method is the better choice as it achieves clearly lower group disagreement in all group type scenarios. The NDCG score results indicate that with AMD, the items in the group recommendation list are in more favorable order for the group members, compared to SADJA and SHA methods.

One additional benefit of AMD is that it can be used without knowledge of the previous recommendation rounds. So, AMD can be expected to provide good results for single-use group recommendations as well as multi-round group recommendations. Further still, the AMD method has an edge over SHA and SADJA methods in the first round

of a recommendation sequence as the dynamic adjusted weights for SADJA and SHA can only by used after the first round whereas AMD method operates with full capacity from the start.

References

[1] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4: 19:1–19:19. *https://doi.org/10.1145/2827872*

[2] Michael J. Pazzani and Daniel Billsus. 2007. Content-Based Recommendation Systems. Springer Berlin Heidelberg, Berlin, Heidelberg, 325–341.

[3] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative Filtering Recommender Systems. Springer Berlin Heidelberg, Berlin, Heidelberg, 291–324.

[4] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A Survey of Collaborative Filtering Techniques. Advances in Artificial Intelligence 2009 (2009). *https://doi.org/10.1155/2009/421425*

[5] Sihem Amer-Yahia, Senjuti Basu Roy, Ashish Chawlat, Gautam Das, and Cong Yu. 2009. Group Recommendation: Semantics and Efficiency. PVLDB 2, 1 (2009), 754–765.

[6] Jae Kyeong Kim, Hyea Kyeong Kim, Hee Young Oh, and Young U. Ryu. 2010. A group recommendation system for online communities. International Journal of Information Management (2010). *https://doi.org/10.1016/j.ijinfomgt.2009.09.006*

[7] Quan Yuan, Gao Cong, and Chin-Yew Lin. 2014. COM: A Generative Model for Group Recommendation. In KDD.

[8] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. ACM Comput. Surv. 51, 4 (2018), 66:1–66:36.

[9] Maria Stratigi, Jyrki Nummenmaa, Evaggelia Pitoura, Kostas Stefanidis. 2020. Fair sequential group recommendations. In: Proceedings of the 35th ACM/SIGAPP Symposium on Applied Computing, SAC 2020

[10] Phonexay Vilakone, Khamphaphone Xinchang, Doo-Soon Park. 2020. Movie Recommendation System Based on Users' Personal Information and Movies Rated Using the Method of k-Clique and Normalized Discounted Cumulative Gain. J Inf Process Syst, Vol.16, No.2, pp. 494-507. *https://doi.org/10.3745/JIPS.04.0169*

[11] Fidel Cacheda, Víctor Carneiro, Diego Fernández, and Vreixo Formoso. 2011. Comparison of Collaborative Filtering Algorithms: Limitations of Current Techniques and Proposals for Scalable, High-performance Recommender Systems. ACM Trans. Web 5, 1 (2011), 2:1–2:33.

[12] Zunping Cheng and Neil Hurley. 2009. Effective diverse and obfuscated attacks on model-based recommender systems. In RecSys.

[13] Kai Yu, A.Schwaighofer, V. Tresp, Xiaowei Xu, and H. Kriegel. 2004. Probabilistic memory-based collaborative filtering. IEEE TKDE 16, 1 (2004), 56–69.

[14] Al-bashiri H, Abdulgabber MA, Romli A, Kahtan H (2018) An improved memory-based collaborative filtering method based on the TOPSIS technique. PLoS ONE 13(10): e0204434. *https://doi.org/10.1371/journal.pone.0204434*

[15] D. Qin, X. Zhou, L. Chen, G. Huang, and Y. Zhang. 2018. Dynamic Connection-based Social Group Recommendation. IEEE Transactions on Knowledge and Data Engineering (2018). *https://doi.org/10.1109/TKDE.2018.2879658*

[16] D. Cao, X. He, L. Miao, G. Xiao, H. Chen and J. Xu. 2021. Social-Enhanced Attentive Group Recommendation. IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 3, pp. 1195-1209. *https://doi.org/10.1109/TKDE.2019.2936475*

[17] Richong Zhang, Han Bao, Hailong Sun, Yanghao Wang, Xudong Liu. 2015. Recommender systems based on ranking performance optimization. Front. Comput. Sci., 2016, 10(2): 270–280 *https//doi.org/10.1007/s11704-015-4584-1*

[18] Francesca Guzzi, Francesco Ricci, and Robin Burke. 2011. Interactive Multi-party Critiquing for Group Recommendation. In Proceedings of the Fifth ACM Conference on Recommender Systems (RecSys ’11). *https://doi.org/10.1145/2043932.2043980*

[19] Dimitris Serbos, Shuyao Qi, Nikos Mamoulis, Evaggelia Pitoura, and Panayiotis Tsaparas. 2017. Fairness in Package-to-Group Recommendations. In WWW.

[20] Ntoutsi E., Stefanidis K., Nørvåg K., Kriegel HP. (2012) Fast Group Recommendations by Applying User Clustering. In: Atzeni P., Cheung D., Ram S. (eds) Conceptual Modeling. ER 2012. Lecture Notes in Computer Science, vol 7532. Springer, Berlin, Heidelberg. *https://doi.org/10.1007/978-3-642-34002-4_10*

[21] Lin Xiao, Zhang Min, Zhang Yongfeng, Gu Zhaoquan, Liu Yiqun , and Ma Shaoping. 2017. Fairness-Aware Group Recommendation with Pareto-Efficiency. In RecSys.

[22] Machado L., Stefanidis K. 2019. Fair team recommendations for multidisclipinary projects. In: WI '19 IEEE/WIC/ACM International Conference on Web Intelligence. *http://dx.doi.org/10.1145/3350546.3352533*

[23] Kaya M., Bridge D., Tintarev N. 2020. Ensuring Fairness in Group Recommendations by Rank-Sensitive Balancing of Relevance. In: RecSys '20: Fourteenth ACM Conference on Recommender Systems. *https://doi.org/10.1145/3383313.3412232*