Investigating and counteracting popularity bias in group recommendations

Article *in* Information Processing & Management · September 2021

DOI: 10.1016/j.ipm.2021.102608

CITATIONS
53

READS
495

2 authors:

Emre Yalcin
Sivas Cumhuriyet University
25 PUBLICATIONS **212** CITATIONS
SEE PROFILE

Alper Bilge
Akdeniz University
45 PUBLICATIONS **1,341** CITATIONS
SEE PROFILE

Investigating and counteracting popularity bias in group recommendations

Emre Yalcin[*,a], Alper Bilge[b]

[a]*Computer Engineering Department, Sivas Cumhuriyet University, 58140 Sivas, Turkey*
[b]*Computer Engineering Department, Akdeniz University, 07058 Antalya, Turkey*

Abstract

Popularity bias is an undesirable phenomenon associated with recommendation algorithms where popular items tend to be suggested over long-tail ones, even if the latter would be of reasonable interest for individuals. Such intrinsic tendencies of the recommenders may lead to producing ranked lists, in which items are not equally covered along the popularity tail. Although some recent studies aim to detect such biases of traditional algorithms and treat their effects on recommendations, the concept of popularity bias remains elusive for group recommender systems. Therefore, in this study, we focus on investigating popularity bias from the view of group recommender systems, which aggregate individual preferences to achieve recommendations for groups of users. We analyze various state-of-the-art aggregation techniques utilized in group recommender systems regarding their bias towards popular items. To counteract possible popularity issues in group recommendations, we adapt a traditional re-ranking approach that weighs items inversely proportional to their popularity within a group. Also, we propose a novel popularity bias mitigation procedure that re-ranks items by incorporating their popularity level and estimated group ratings in two distinct strategies. The first one aims to penalize popular items during the aggregation process highly and avoids bias better, while the second one puts more emphasis on group ratings than popularity and achieves a more balanced performance regarding conflicting goals of mitigating bias and boosting accuracy. Experiments performed on two real-world benchmark datasets demonstrate that both strategies are more efficient than the adapted approach, and empowering aggregation techniques with one of these strategies significantly decreases their bias towards popular items while maintaining reasonable ranking accuracy.

Key words: Group recommendation, popularity bias, aggregation techniques, ranking-based recommendation.

1. Introduction

With recent advances in the Web technology, individuals are getting more encouraged to utilize digital platforms for performing their daily routines such as listening to music[1], booking a

[*]Corresponding author
Email addresses: eyalcin@cumhuriyet.edu.tr (Emre Yalcin), abilge@akdeniz.edu.tr (Alper Bilge)
[1]https://www.spotify.com/

hotel[2], shopping[3], and watching movies[4]. However, users are usually not able to find relevant products/services due to a vast and continuously growing amount of options in such systems, which is referred to as the *information overload* problem (Bawden and Robinson, 2020). Recommender systems (RSs) focus mainly on discovering engaging content by filtering out irrelevant data (Shambour, 2021). Thus, they improve the usability and attraction of the system by supporting the decision-making process of users.

Traditional RSs aim to satisfy individual users by producing referrals reflecting their tastes and interests. Nevertheless, individuals are inclined to experience particular activities with familiar people (e.g., traveling with friends (Nguyen and Ricci, 2017), watching movies with family members (Agarwal et al., 2017)) or compelled to act together with a community in some cases such as commuting by public transport (Van Lierop et al., 2018). Group recommender systems (GRSs) have been introduced as more complicated recommendation mechanisms for providing recommendations satisfying communities rather than individual users and recently utilized in many different application domains such as music (Kim and El Saddik, 2015), movies (Castro et al., 2017; Yalcin et al., 2021), touristic activities (Christensen et al., 2016), and so on. These systems provide group recommendations based on the group profiles reflecting the group members' tendencies and characteristics as a whole. In general, such group profiles are constructed by utilizing various algebraic approaches, termed aggregation techniques, combining group members' preferences or individual recommendations Masthoff (2015); Seo et al. (2018). Such techniques are crucial for estimating accurate group preferences and directly influence the provided group recommendations' quality.

Recommendation algorithms are intrinsically trained on user preferences distributed nonuniformly among items, as a high number of users evaluate particular items while other items receive only a few ratings. Such phenomenon orients utilized algorithms to feature popular items more than niche (i.e., non-popular) ones (Park and Tuzhilin, 2008; Abdollahpouri et al., 2017). Thus, favorite items gain more visibility in the produced recommendation lists, which further boosts their consumption rate by individuals (Boratto et al., 2021). On the other hand, niche items involved in the long-tail, especially those recently included in the system, cannot get the deserved attention, even when they would be of reasonable interest. Such inherent prejudice of recommender systems is also referred to as the popularity bias problem and leads to *"the rich get richer"* impact in favor of a few popular items (Kamishima et al., 2014). In recent years, scrutinizing the issues of popularity bias caused by recommendation algorithms (Jannach et al., 2016; Abdollahpouri et al., 2019b; Boratto et al., 2019) and mitigating its adverse effects on recommendations (Kamishima et al., 2014; Boratto et al., 2021) have been getting more attention.

Although a few GRSs employ pre-defined user groups, it is a fact that groups of users having similar tastes are generally automatically identified using a traditional clustering algorithm (Boratto and Carta, 2015; Yalcin and Bilge, 2021). Since such clustering algorithms calculate similarities among users by considering the set of co-rated items, only the users submitting votes for the same items are discovered to be similar; hence, they are assigned into the same group. Such group construction method leads to having user groups where rating counts become imbalanced towards co-rated items that orient aggregation techniques become prone to feature such items in group recommendations, leading to the popularity bias problem. Also, respected group

[2]https://www.booking.com/
[3]https://www.ebay.com/
[4]https://www.netfix.com/

members usually influence other individuals in decision-making, as groups consist of users having similar interests and social relationships, by its nature (Barzegar Nozari and Koohi, 2020). Therefore popular items become more and more consumed by group members, as individuals can be easily manipulated by other members' opinions on favored items. Thus, it can be concluded that GRSs are inherently vulnerable against popularity bias issues as they operate on a smaller and highly similar set of users compared to traditional recommenders. However, the popularity bias concept, its effects on GRS outputs, and how to mitigate these effects are not considered in the literature.

Keeping in mind that the aggregation techniques are the fundamental components in GRSs (Seo et al., 2018; Yalcin et al., 2021), popularity bias problems originate from group ratings estimated by these techniques. Therefore, in this study, we examine prominent traditional aggregation techniques regarding the popularity bias caused by them and propose novel mitigation methods for possible popularity issues in group recommendation scenarios. The main contributions of this study are summarized in the following.

1. We provide a comprehensive analysis of standard aggregation techniques in terms of the level of the popularity bias they may lead to in produced group recommendation lists. To this end, we consider the most prominent eight aggregation techniques to investigate their effects on popularity bias.
2. We adopt an efficient popularity-debiasing approach developed for conventional recommendation algorithms (Abdollahpouri et al., 2018) and propose its adapted version suitable for group recommendation scenarios. More specifically, this adapted method incorporates each item's popularity rate and the calculated group ratings for them in a straightforward way to re-rank the produced recommendation lists for groups.
3. We propose two enhanced re-ranking procedures to overcome some shortcomings of the adapted method in the combination of the popularity rates and the group ratings. These methods follow more robust combination strategies to mitigate the popularity bias effects in aggregation and improve the quality of the group recommendations.

The rest of the study is organized as follows: The next section explains the baseline aggregation techniques used for group recommendations and analyzes them according to their combining strategies. Section 3 presents a literature summary of well-known GRSs and some prominent studies related to popularity bias in traditional recommender systems. Section 4 introduces both the modified approach and the proposed two popularity-debiasing procedures for group recommendations, and the following section presents experimental studies, including obtained results and gained insights. Finally, Section 6 concludes the paper and gives information about our future research directions.

2. Background on utilized aggregation techniques

GRSs are enhanced recommendation mechanisms suggesting appropriate items for groups of users sharing similar interests, rather than individuals (Masthoff, 2015). Also, it is common to utilize group recommendation strategies to overcome potential cost or context problems that may appear in individual recommendations (Baltrunas et al., 2010; Boratto and Carta, 2015). Here, the main aim is to consider a set of users having similar interests as a whole and provide group recommendations for them, rather than attempting to generate recommendations individually (Seo et al., 2018; Yalcin et al., 2021). Such group recommendations are generally provided based

on the group profiles generated by combining the group members' preferences to end up having group preferences on items. This task is commonly accomplished utilizing a proper aggregation approach (Seo et al., 2018).

Although various modern aggregation approaches have been developed in recent years, they often rely on some traditional aggregation techniques due to their efficiency (Yalcin et al., 2021). In other words, such traditional techniques are usually utilized as the core of these approaches to estimate convenient group preferences and produce qualified group recommendations. More specifically, these techniques are typically classified into three main categories depending on their combining strategies, which are explained in detail below (Felfernig et al., 2018). We consider the most prominent aggregation techniques for each strategy in this study to examine how the combining strategy leads to popularity bias in group recommendations. To illustrate how these techniques work, we also provide an example user-item matrix in Table 2, including preferences of a group with four members for five items on a ten-star rating scale. Here, $\perp$ indicates the unrated items by the members.

	i_1	i_2	i_3	i_4	i_5
u_1	$\perp$	8	$\perp$	3	9
u_2	$\perp$	6	8	$\perp$	3
u_3	2	7	$\perp$	$\perp$	3
u_4	4	$\perp$	$\perp$	5	6

Table 1: A user-item matrix to illustrate the aggregation techniques

Consensus-based strategy: The aggregation techniques following a *consensus-based* strategy determine group ratings by taking into account the preferences of all members in the group, which leads to providing a consensus among members. Such techniques are effective and easy to implement, as they utilize basic arithmetic operations such as averaging, multiplication, and addition. The most popular examples using this strategy are Average (Avg) (Wang et al., 2019; Barzegar Nozari and Koohi, 2020; Zhiwen et al., 2005), Multiplicative (Mul) (Christensen and Schiaffino, 2011), Additive Utilitarian (AU) (Yalcin et al., 2021; McCarthy, 2002; Boratto et al., 2016), and Average without Misery (AwM) (Chao et al., 2005; Yalcin and Bilge, 2020).

More specifically, the Avg determines a group rating for an item by estimating the average of group members' preferences for the corresponding item. On the other hand, the Mul calculates individual ratings' multiplication, while the AU sum up them to achieve group ratings. As a variant of the Avg technique, the AwM does not calculate an average score for the items received a rating below a pre-defined threshold from at least one group member; thus, it evaluates such items as not recommendable, disregarding them in aggregating. According to the user-item matrix given in Table 1, group ratings estimated by these techniques are presented in Table 2. Here, the threshold value for the AwM technique is selected as 6.

Majority-based strategy: The aggregation techniques using a *majority-based* strategy focus on featuring popular items among group members. The most prominent example of *majority-based* techniques are Approval Voting (AV) (Seo et al., 2018; Yalcin et al., 2021; Boratto et al., 2016) and Borda Count (BC) (Felfernig et al., 2018). More specifically, the AV

	i_1	i_2	i_3	i_4	i_5
Avg	3	6	8	4	5.25
Mul	8	336	8	15	486
AU	6	21	8	8	21
AwM	-	7	8	-	-

Table 2: Group ratings by the *Avg, Mul, AU* and *AwM* techniques

technique determines the group rating for an item by counting how many times it is rated with a score above a pre-defined threshold from members of the group. In other words, it disregards the ratings that are below the threshold during the aggregation process. To grasp how the AV technique works, we present an example in Table 3, where the threshold value is selected as 6.

	i_1	i_2	i_3	i_4	i_5
AV	0	2	1	0	1

Table 3: Group ratings by the *AV* technique

On the other hand, the BC performs the aggregation process by considering the items' rankings rather than the number of votes they received. Concretely, this technique first scores the items based on each member's ranking results in the group. (i.e., items get a ranking score in ascending order where the item having the lowest rating gets the rank of 0). Then, it sums up such assigned scores to achieve ultimate group ratings for items. We provide an example in Table 4 to show how UL works in practice based on the ratings given in Table 1.

	i_1	i_2	i_3	i_4	i_5
u_1	$\perp$	1	$\perp$	0	2
u_2	$\perp$	1	2	$\perp$	0
u_3	0	2	$\perp$	$\perp$	1
u_4	0	$\perp$	$\perp$	1	2
BC	0	4	2	1	5

Table 4: Group ratings by the *BC* technique

Borderline strategy: It addresses the aggregation techniques that consider only a subset of ratings provided by members of the group. The most well-known techniques following this strategy are Least Misery (LM) (O'Connor et al., 2001; Christensen and Schiaffino, 2011) and Most Pleasure (MP) (Ahmad et al., 2017; Boratto et al., 2016). Specifically, the LM selects the lowest rating given by members in the group, while the MP selects the highest rating to reach group ratings. Table 5 presents group ratings estimated by the LM and MP techniques according to the preferences given in Table 1.

	i_1	i_2	i_3	i_4	i_5
LM	2	6	8	3	3
MP	4	8	8	5	9

Table 5: Group ratings by the *LM* and *MP* technique

3. Related work

This study builds on and relates to literature from GRSs and popularity bias in personalized recommendations; therefore, we divide this section into two parts reviewing some prominent GRSs and introducing existing researches that consider issues of popularity bias.

3.1. Group recommender systems (GRSs)

Over the past twenty years, many GRSs have been proposed for different scenarios in various application domains such as movies (O'Connor et al., 2001; Quijano-Sanchez et al., 2011; Barzegar Nozari and Koohi, 2020; Castro et al., 2018), music (McCarthy and Anagnost, 2000; Crossen et al., 2002; Christensen and Schiaffino, 2011; Chao et al., 2005; Zhiwen et al., 2005), touristic attractions (Ardissono et al., 2003; Jameson, 2004; McCarthy et al., 2006), and restaurants (McCarthy, 2002). Since there is no single aggregation technique that achieves superior performance in all application scenarios, the aggregation techniques utilized in such systems to generate group recommendations usually differ (Seo et al., 2018).

PolyLens (O'Connor et al., 2001) is introduced as an extension of the well-known MovieLens to provide movie recommendations to user groups instead of individual users. This system employs the LM technique to aggregate individual preferences for movies. Another example of GRS in the domain of movie is HappyMovie (Quijano-Sanchez et al., 2011), which is developed as a Facebook application and utilizes the Avg and LM as aggregation techniques. More specifically, it produces recommendations for users' groups by considering trust among users in the group and their interests in the domain. The NNMG (Castro et al., 2018) is another approach providing movie recommendations for user groups, where the AVG and LM techniques are used for combining both produced recommendations and individual preferences. This system aims to find out the natural noise existing in group members' preferences and eliminate its adverse effects on the recommendations. Also, IBGR (Barzegar Nozari and Koohi, 2020) and TruGRC (Wang et al., 2019) are enhanced GRSs providing movie recommendations for groups by utilizing the Avg during the aggregation process. More specifically, the IBGR focuses on determining group members' influence on each other by considering trust and similarity among members and then utilize them to weight individual preferences while constructing the group profiles. Like the IBGR, the TruGRC considers the social relationships based on trust among members and incorporates them into the aggregation process.

MusicFX (McCarthy and Anagnost, 2000) is a smart platform that picks the radio station for exercise music for a group of people at a gym using the probabilities of favorite music genres of users as a whole. Such probabilities are estimated using the AwM technique; thus, it considers only genres where all user preferences are above a threshold. Another popular GRS in the music domain is FlyTrap (Crossen et al., 2002) that produces a virtual DJ to recommend a playlist for users in a particular room. To this end, the virtual DJ exposes users' musical preferences and combines them with some background knowledge, such as the transitions between songs and interrelation between music genres. This combination is performed by utilizing a variant of the AV

technique. Also, jMovieGroupRecommender and jMusicGroupRecommender (Christensen and Schiaffino, 2011) are introduced as the entertainment GRSs to provide group recommendations for movies and music, respectively. Both of them include several recommendation approaches where Avg, Mul, and LM are utilized in harmony. Finally, Adaptive In-Vehicle Multimedia System (Zhiwen et al., 2005) and Adaptive Radio (Chao et al., 2005) are other salient examples of GRSs in the domain of music, and the Avg technique is opted for constructing group profiles in both of them.

Pocket Restaurant Finder (McCarthy, 2002) is a GRS recommending a sequence of restaurants for user groups. In this system, individuals' preferences for location and food (e.g., the type of cuisine, price, and taste) are combined by utilizing the AU technique. Also INTRIGUE (Ardissono et al., 2003), TDF (Jameson, 2004), and CATS (Ardissono et al., 2003) are some prominent GRSs in the context of tourism activities. Specifically, INTRIGUE produces a list of suggested touristic attractions for guided tours based on tour participants' characteristics in favor of disabled and children. On the other hand, TDF supports the decision-making process of a group of people planning to go on a holiday together by recommending an attractive destination. CATS suggests best-suited ski-packages for a group of people by considering their demands and needs. Finally, all of these systems, INTRIGUE, TDF, and CATS, utilize the Avg as the aggregation technique to construct group profiles for producing group recommendations.

As can be seen from the present section, various GRSs are developed for different purposes; however, the utilized aggregation mechanisms are usually built on top of traditional aggregation techniques, which are explained in detail in the previous section. Therefore, such techniques are considered as vital components for GRSs in providing group recommendations.

3.2. Popularity bias in personalized recommendations

Traditional recommender systems commonly suffer from the famous popularity bias problem, which is occurred by the tendency of utilized algorithms on featuring popular items more than disfavored ones (Cañamares and Castells, 2018). The effects of such a phenomenon on personalized recommendations have been of interest to recent studies in recommender systems. In this sense, the existing literature has mainly focused on analyzing the level of popularity bias caused by traditional algorithms for different application domains such as movies (Abdollahpouri et al., 2019b,c), music (Jannach et al., 2016; Kowald et al., 2020), tourism activities (Sánchez, 2019) and online education (Boratto et al., 2019), and developing practical solutions treating its adverse effects in the recommendations. The proposed approaches for mitigating popularity bias in personalized recommendations are usually classified into the three main categories, i.e., *pre-processing, in-processing,* and *post-processing*, based on how they are involved in the recommendation process (Boratto et al., 2021).

More specifically, *pre-processing* approaches aim to reduce the imbalances in rating distributions by altering the data on which algorithms are trained. For example, in (Park and Tuzhilin, 2008), the item set is divided into two parts, named as tail and head, and the tail ones are clustered based on the number of received ratings. Then, the recommendations for tail items are produced using only ratings in the corresponding cluster, while for head ones are estimated based on the ratings for individual items. The work in (Jannach et al., 2015) counteracts the popularity bias by utilizing input user-item tuples, where the observed items are more popular than the unobserved ones. Finally, (Chen et al., 2018) propose featuring the unobserved items by using a probability distribution based on popularity.

In-processing approaches, on the other hand, focus on modifying standard algorithms to simultaneously consider popularity and relevance, using specific constraints or performing a joint

optimization. For instance, (Oh et al., 2011) propose a method that recommends tail items by evaluating the personal popularity tendency of individuals. Such tendencies are used to diversify recommendations by reasonably penalizing favored items in the recommendation process. (Kamishima et al., 2014) aim to improve the statistical independence between produced personalized recommendations and their popularity. Also, (Abdollahpouri et al., 2017) propose the RankALS optimization algorithm that generates recommendation lists where ranking accuracy is balanced with intra-list diversity. (Hou et al., 2018) introduce a framework for personalized recommendations, which initially determine familiar neighbors between two items with given popularity, and then prune popular ones to obtain a balanced common-neighbor similarity index. Finally, (Boratto et al., 2021) present a popularity debiasing approach that minimizes the correlation item popularity and user-item relevance; thus, items are treated more equally along the popularity tail.

Finally, the main goal in *post-processing* approaches is to re-rank a recommended list or create a new one according to a particular constraint. As an example, (Abdollahpouri et al., 2018) present an approach that estimates ultimate scores by weighting the predicted ratings by inversely proportional to the item popularity and then utilizes them to re-rank the recommendation list. Similarly, (Abdollahpouri et al., 2019a) introduce the *xQuad* re-ranking algorithm that balances the trade-off the coverage of long-tail items and ranking accuracy. Finally, (Jannach et al., 2015) utilize user-specific weights that are useful in balancing popularity and ranking accuracy.

Although many recent studies aim to analyze popularity bias issues and counteract its effects in personalized recommendations, to the best of our knowledge, the popularity bias concept is not previously considered in the context of GRSs. Since the aggregation techniques play a vital role in producing group recommendations, there is a need to examine them in terms of popularity bias and develop novel bias treatment methods adapted to group recommendations.

4. The proposed popularity-debiasing methods for group recommendations

This section introduces an efficient popularity-debiasing method for personalized recommendations and its adapted version for group recommendation scenarios. Then, we present two novel approaches that combine the popularity level of items and group ratings estimated for them in two different ways to mitigate popularity bias effects in group recommendations.

4.1. The Adapted Value-aware Ranking method

A common practice for counteracting popularity bias in personalized recommendations is to feature long-tail items by penalizing the popular ones during the recommendation process. The *Value-aware Ranking (VaR)* (Abdollahpouri et al., 2018) is an efficient debiasing approach following this practice by re-ranking items based on a weighting strategy. More specifically, the *VaR* approach initially assigns higher weights for the long-tail items while gives lesser weights to the favored ones to counterbalance bias induced by their popularity. Such weights in this approach are estimated using the formula given in Eq. 1.

$$w_i = \frac{1}{log_2(p(i))} \tag{1}$$

where $p(i)$ indicates the number of times that item i has been rated by the users. Note that the *VaR* utilizes the log function at the denominator to limit the range of $p(i)$ value; hence, popular items are not penalized harshly.

After determining weights for each item, the *VaR* computes a predicted rating $\hat{r}_{u,i}$ for user u on item i using any traditional collaborative filtering algorithm such as SVD++ (Koren, 2008), item-based or user-based kNN (Herlocker et al., 1999). Finally, it estimates a value-aware ranking score $\Upsilon_{u,i}$ for user-item pair (u, i) by combining the predicted rating and the calculated weight of the corresponding item, as formulated in Eq. 2. These ranking scores are then used to recommend a ranked list of items for each user after sorting.

$$\Upsilon_{u,i} = \alpha w_i + (1 - \alpha)\hat{r}_{u,i} \tag{2}$$

where α is an adjusting parameter that controls the significance level of each of the components in Eq 2. Specifically, smaller α value means more weight for predicted ratings for items by less considering their popularity; higher α value, on the other hand, leads to penalizing popular items more and shifting the balance towards the long-tail ones.

Although such a combination of the predicted ratings and the calculated item weights seems like a reasonable approach, utilizing their weighted average to achieve ranking scores raises a fundamental problem: the item weights calculated by Eq. 1 vary from 0 to 1, while the predicted ratings usually vary in a broader range, depending on the utilized rating scale. Such inconsistency between weights and estimated ratings results in an incompatibility issue, especially in large-scale rating systems, which leads to a dramatic decrease in the impacts of the item weights on the ultimate ranking scores. To handle this problem, we introduce an adapted version of the *VaR* approach, termed the *Adapted Value-aware Ranking (AdaptedVaR)*, which is also applicable for group recommendation scenarios.

The *AdaptedVaR* first calculates item weights to be used in estimating ranking scores, as in the *Var*, but differ in that it estimates these weights for each user group separately. In other words, this adapted version considers only ratings given by the members of the group when determining item weights for the corresponding group. Concretely, if g indicates a particular group of users, then the weight value $w_{g,i}$ of the item i for the group g is calculated as given in Eq. 3

$$w_{g,i} = \frac{1}{log_2(p_g(i))} \tag{3}$$

where $p_g(i)$ denotes the number of times that i has been rated by members of g.

Since ranked recommendation lists for user groups are generally produced by sorting group ratings estimated using a proper aggregation approach, such group ratings stand for the predicted ratings utilized in the *VaR* method. Therefore, after calculating item weights for g, the *Adapted-VaR* determines a group rating $R_{g,i}$ for g on i using one of the traditional aggregation techniques explained in Section 2. Different from *VaR* approach, *AdaptedVaR* performs a normalization process to make the values of $R_{g,i}$ and $w_{g,i}$ compatible before computing ultimate ranking scores. For this purpose, it first transforms the calculated group ratings into [0, 1] interval using min-max normalization, and then combine them into having ranking scores, as in Eq. 4.

$$\Upsilon_{g,i} = \alpha w_{g,i} + (1 - \alpha)\overline{R_{g,i}} \tag{4}$$

where $\Upsilon_{g,i}$ is the value-aware ranking score for group g on item i, and $\overline{R_{g,i}}$ is the normalized group rating of group g on item i.

4.2. The Enhanced Re-ranking Procedures for group recommendations

The previous section presented a straightforward popularity-debiasing method for personalized recommendations and an adapted version suitable for group recommendation scenarios.

The adapted version calculates a weight for each item based on their popularity in the group; and then combines them with normalized group ratings to achieve ranking scores that will be used for recommending top-N items to groups. This combination is performed by using a weighted average strategy, where the weight parameter, i.e., α, is randomly selected from the set of $\{0, 0.1, \cdots, 0.9, 1\}$, as given in Eq. 4. However, utilizing such a random adjusting mechanism makes it challenging to define the optimal α value for providing high-quality group recommendations where the trade-off between ranking accuracy and long-tail item coverage is balanced. Furthermore, when an item is rated by only one group member, the w value computed by the formula given in Eq. 3 yields an undefined value, which leads to a difficulty in re-ranking items. To overcome such limitations, we propose a novel popularity-debiasing procedure for group recommendations, termed *Enhanced Re-ranking Procedure (ERP)*, that integrates group ratings and item weights utilizing two different robust ways.

As in the *AdaptedVaR* approach, the *ERP* also initially follows an item weighting mechanism to determine items' popularity level among group members; however, the utilized weighting method in the *ERP* focuses on more penalizing the favored items compared to that in the *AdaptedVaR*. For this purpose, the *ERP* first assigns a weight for each item by counting the number of times that the corresponding item is rated by group members; it then normalizes the weights so that each of them lies in [0, 1] by performing min-max normalization. Finally, it calculates final item weights using the formula given in Eq. 5.

$$w_{g,i} = 1 - \overline{p_g(i)} \tag{5}$$

where $\overline{p_g(i)}$ is the normalized value of number of users who rated item i in group g.

Having the final item weights are computed, the *ERP* determines group ratings for items using one of the appropriate traditional aggregation techniques explained in Section 2. Also, depending on the utilized aggregation technique, the estimated group ratings vary in a highly wider range than that of item weights, which requires a normalization process before combining to obtain compatible values. For this purpose, we again perform min-max normalization to transform calculated group ratings into [0, 1] scale. At the final step, the *ERP* combines group ratings and item weights to achieve ultimate ranking scores that will later be used to recommend ranked lists of items to groups, utilizing two different approaches: *Multiplicative* and *Augmentative*. Both approaches are inspired by previously introduced practical methods to properly combine two related variables in the domain of RSs (Vozalis and Margaritis, 2004, 2006; Yalcin and Bilge, 2021). Also, both *ERP* variants are the *post-processing* approaches; therefore, they can be easily applied to any aggregation technique producing group ratings for items. The following describes how the ranking scores are computed using these methods.

Multiplicative (Mul): This approach uses the item weights as a factoring coefficient on the calculated group ratings, which leads to more penalizing favored items in the group and featuring long-tail ones during the recommendation process. Concretely, *Multiplicative* approach multiplies the normalized group rating $\overline{R_{g,i}}$ estimated for item i with the weight $w_{g,i}$ calculated for i, to achieve a ranking score $\Upsilon_{g,i}$ for i in the group g, as in formulated Eq. 6. Note that we shall denote this approach as the ERP_{Mul} for the rest of the study.

$$\Upsilon_{g,i} = w_{g,i} \times \overline{R_{g,i}} \tag{6}$$

Augmentative (Aug): This approach employs the group ratings as the driving force, while the item weights as an additive factor on the ultimate ranking scores, and formulated in Eq.

7. In other words, the *Augmentative* approach values group ratings as the decisive factor, while it utilizes item weights as an auxiliary influencer. Note that this approach is called the ERP_{Aug} for the rest of the study.

$$\Upsilon_{g,i} = \overline{R_{g,i}} + (\overline{R_{g,i}} \times w_{g,i}) \tag{7}$$

4.3. Illustrative example

To clarify how the *AdaptedVaR* and our *ERP* variants (i.e., ERP_{Mul} and ERP_{Aug}) determine the ultimate ranking scores (Υ) for items, we provide an illustrative example including each step of these methods in Table 6 based on the ratings given in Table 1. In this toy example, we utilize the Avg as the aggregation technique to achieve the group ratings (R). Note also that the value of α is selected as 0.5 when performing the *AdaptedVaR* method.

Method	*Step*	i_1	i_2	i_3	i_4	i_5	*Explanation*
AdaptedVaR	w	1	0.63	NaN	1	0.5	*using Eq. 3*
	R	3	6	8	4	5.25	*using Avg technique*
	$\overline{R}$	0	0.60	1	0.20	0.45	*by min-max normalization*
	Υ	**0.50**	**0.62**	***N/A***	**0.60**	**0.48**	*using Eq. 4*
ERP_{Mul}	w	0.33	0.67	0	0.33	1	*using Eq. 5*
	R	3	6	8	4	5.25	*using Avg technique*
	$\overline{R}$	0	0.60	1	0.20	0.45	*by min-max normalization*
	Υ	**0**	**0.40**	**0**	**0.06**	**0.45**	*using Eq. 6*
ERP_{Aug}	w	0.33	0.67	0	0.33	1	*using Eq. 5*
	R	3	6	8	4	5.25	*using Avg technique*
	$\overline{R}$	0	0.60	1	0.20	0.45	*by min-max normalization*
	Υ	**0**	**1**	**1**	**0.26**	**0.90**	*using Eq. 7*

Table 6: The steps of the *AdaptedVaR*, ERP_{Mul}, and ERP_{Aug} methods, in calculating ultimate ranking scores

Although the *AdaptedVaR*, ERP_{Mul} and ERP_{Aug} seem to be methods following similar strategies in computing ranking scores for items, they produce recommendation lists where the sorting of the items highly differs, as can be seen from Table 6. For example, *AdaptedVaR* values the i_2 as the most recommendable item, while the ERP_{Mul} recommends i_5 at the top of the its produced ranked list. On the other hand, the best-ranking score for the ERP_{Aug} method is observed for both i_2 and i_3. Also, the *AdaptedVaR* cannot calculate a ranking score for i_3, as can be followed by Table 6. The main reason for this observation is that when an item is rated by only one group member, the weight calculated using the formula given in Eq. 3 yields an undefined value. Such a shortcoming of the *AdaptedVaR* can also be considered another evidence of our *ERP* variants' robustness against the *AdaptedVaR* method.

5. Experimental studies

In this section, we present various experiments on real-world datasets to examine the effectiveness of the proposed popularity-debiasing methods for group recommendations in terms of ranking accuracy and popularity bias, and then discuss the insights gained from the empirical findings.

5.1. Datasets and evaluation metrics

In the experiments, we employ two different versions of the famous MovieLens dataset, i.e., MovieLens100K (MLP) and MovieLens1M (MLM), collected and released by the GroupLens research team at the University of Minnesota[5]. MLP and MLM datasets consist of user preferences on movies represented by discrete ratings on a 5-star rating scale. Also, we utilize the Ciao dataset that is crawled from a real-world social media platform[6] and includes user ratings provided for products. Because the original Ciao dataset is too large and extremely sparse, we make use of a subset of the collection where each item and user have at least 20 ratings, also referred to as Ciao20. Table 7 presents detailed information about the MLP, MLM, and Ciao20 datasets.

Dataset	#Users	#Items	#Ratings	Sparsity (%)	Rating Scale
MovieLens100K (MLP)	943	1,682	100,000	93.70	5-star, {1,5}
MovieLens1M (MLM)	6,040	3,952	1,000,209	95.75	5-star, {1,5}
Ciao20	3,278	1,351	52,717	98.81	5-star, {1,5}

Table 7: Detailed information of the MLP, MLM, and Ciao20 datasets

Since the MLP, MLM, and Ciao20 consist of only the user preferences individually provided, they do not include any information about the user groups. Therefore, for these datasets, we utilize groups of similar users that are artificially generated, as explained in the following section. However, to scrutinize our proposed methods' performance comprehensively, we also employ an additional dataset, namely CAMRa2011, which is released from the competition of context-aware movie recommendation[7]. This dataset consists of movie rating records for both individual users and pre-defined user groups, which enables us to scrutinize outcomes in real-world group formation scenarios. Properties of the CAMRa2011 dataset are presented in Table 8.

#Users	602
#Items	7,710
#Groups	290
#Avg. Group Size	2.08
#User-Item Ratings	116,344
#Group-Item Ratings	145,068
Rating Scale	21-star, {0,100}

Table 8: Properties of the CAMRa2011 dataset

To investigate the proposed approaches' accuracy performance on top-N group recommendations, we utilize the normalized Discounted Cumulative Gain (nDCG) metric, commonly used in studies on group recommendations (Seo et al., 2018). More specifically, the nDCG metric measures the quality degree of a ranked recommendation list, considering both the items' positions in the recommended list and their actual ratings. Suppose that u is a member of group g,

[5] http://www.grouplens.org/
[6] https://www.ciao.co.uk/
[7] https://2011.camrachallenge.com/2011/

and $r_{u,i}$ indicates the actual rating of u for item i. Also assume that $\{i_1, i_2, \cdots, i_N\}$ denotes the ranked list of N-items recommended to the g. Then, Discounted Cumulative Gain (DCG) and nDCG for each user in the g are computed using the formulas given in Eqs. 8 and 9, respectively.

$$\mathrm{DCG}_N^u = r_{u,i_1} + \sum_{n=2}^{N} \frac{r_{u,i_n}}{log_2(n)} \tag{8}$$

$$n\mathrm{DCG}_N^u = \frac{\mathrm{DCG}_N^u}{\mathrm{IDCG}_N^u} \tag{9}$$

where IDCG_N^u demonstrates the maximum possible gain, which can be calculated by re-ordering the recommended N items to be in the ideal order for u.

In calculating the nDCG metric, it is required to know the group members' actual ratings for each item in the recommended list. However, it is almost impossible to have all users' actual votes since the datasets used in recommender systems are generally sparse, making it challenging to compute nDCG scores. To cope with this problem, we predict users' actual ratings by utilizing SVD++ (Koren, 2008), which is a modern collaborative filtering algorithm, as in (Yalcin et al., 2021; Sacharidis, 2019). In this way, we obtain ground truths for each group member, which will be utilized in computing nDCG scores of produced top-N group recommendations.

In the evaluation process, we also employ the Group Average Popularity (GAP) metric, which is recently developed to estimate the popularity bias of recommended items for different user groups instead of individuals (Abdollahpouri et al., 2019c; Kowald et al., 2020). More specifically, the GAP metric relies on two specific measurements denoted as the $GAP_p(g)$ and the $GAP_r(g)$. The former measures the average popularity of items in the profiles (p) of users in the group g, as formulated in Eq. 10. On the other hand, the latter measures the average popularity of the items recommended to g using a suitable aggregation technique, as formulated in Eq. 11.

$$GAP_p(g) = \frac{\sum_{u \in g} \frac{\sum_{i \in p_u} \Phi(i)}{|p_u|}}{|g|} \tag{10}$$

$$GAP_r(g) = \frac{\sum_{u \in g} \frac{\sum_{i \in r_u} \Phi(i)}{|r_u|}}{|g|} \tag{11}$$

where p_u indicates the list of items in the profile of user u, and r_u denotes the list of recommended items to group g of which user u belongs to. Also, $\Phi(i)$ is the popularity of item i, computed by dividing the number of times it is rated in the group by the number of users in g.

Finally, this metric calculates the change in GAP values, i.e., ΔGAP, showing the quantity of undesirable popularity in the top-N group recommendations imposed by the utilized aggregation technique to g, as given in Eq. 12. Note that higher ΔGAP values demonstrate amplification of popularity bias by utilized aggregation technique; smaller ΔGAP values, on the other hand, are obtained when the recommendations are less concentrated on popular items than the profile of the users.

$$\Delta GAP = \frac{GAP_r(g) - GAP_p(g)}{GAP_p(g)} \tag{12}$$

5.2. Experimentation methodology

Although a few GRSs employ established groups, sets of users with similar interests are not usually predefined; therefore, the initial procedure of most of the existing studies on group recommendations is to divide users into groups automatically, which is referred to as automatic identification of groups (Boratto et al., 2010). The most prominent approach for performing this task is *Predict&Cluster* (Boratto and Carta, 2014; Boratto et al., 2016), identifying groups of similar users by applying k-means clustering algorithm not on the original user-item matrix containing only user ratings, but on the full matrix where missing ratings are predicted using an appropriate collaborative filtering algorithm. Therefore, to identify user groups in this study, we follow the *Predict&Cluster* strategy, where missing ratings are predicted using SVD++, which is one of the most efficient collaborative filtering algorithms (Koren, 2008). Note that such artificially generated user groups are utilized for only the MLP, MLM, and Ciao20 datasets, as the information about the groups is available on the CAMRa2011 dataset.

In the experiments, we perform a five-fold cross-validation experimentation methodology to evaluate the proposed approaches. To this end, we randomly partition the item set into five subsets; thus, each subset consists of 20% of the items. At each iteration, one of the constructed subsets is utilized as the test set, and the combination of the remaining four subsets is used as the training set. The training set is employed to identify user groups using the *Predict&Cluster* strategy explained above. On the other hand, the test set is used to examine the performance of aggregation techniques and their modified versions with popularity-debiasing methods.

To investigate the influence of group size on the proposed popularity-debiasing methods, when applying the k-means algorithm for constructing user groups, we consider the groups of three different sizes (P), which are around 10 (i.e., small groups), 50 (i.e., medium groups), and 100 (i.e., large groups). In doing so, we estimate the number of groups (k) for the k-means algorithm with the ratio of the total number of users in the dataset to the P value which stands for the average size of the groups.

After user groups are identified, we estimate group ratings for items by utilizing the aggregation techniques: Avg, Mul, AU, AwM, AV, LM, MP, and BC, ensuring that each aggregation strategy previously explained in Section 2 is represented in the experiments. We also employ the modified version of each aggregation technique with the proposed *AdaptedVaR*, ERP_{Mul}, and ERP_{Aug} popularity-debiasing methods, in computing the group ratings. Note that the threshold value for both AV and AwM techniques is selected as 3 for MLP, MLM, and Ciao20 datasets, as the positive ratings usually correspond to 4 and 5 for a 5-star rating scale (Bobadilla et al., 2010). On the other hand, the user preferences in the CAMRa2011 dataset are on the scale of 21-star and vary from 0 to 100. Therefore, this dataset's threshold value is intuitively selected as 60, which is equivalent to the rating of 3 when the user preferences are normalized into [1-5] rating scale. Also, we consider three different α values, i.e., 0.25, 0.5, and 0.75, while performing the *AdaptedVaR* method to see its effects on the ranking accuracy and popularity-bias.

After group ratings are computed, we sort them in descending order; and then select top-N items as a list of group recommendations, where N is set as 5. At the final step, to evaluate the quality of produced group recommendations in terms of ranking accuracy and popularity-bias, we calculate nDCG and ΔGAP scores for each group and average them. Note that we separately compute nDCG scores for each group member and take their average to achieve the ultimate nDCG score of the group. Finally, we take the average of nDCG scores observed for all groups to achieve ultimate accuracy performances.

5.3. Experimental results

To scrutinize the performance of the proposed popularity-debiasing procedures on producing group recommendations, we conducted an extensive set of experiments with various parameters, including the utilized aggregation technique and the group size (P). In these experiments, we compared the eight aggregation techniques with their modified versions by the proposed bias treatment methods (i.e., *AdaptedVaR*, ERP_{Mul}, and ERP_{Aug}) in terms of their ΔGAP and nDCG performances for the MLP, MLM, Ciao20, and CAMRa2011 datasets, as presented in Tables 9, 10, 11, and 12, respectively. In the presented results, we consider the average of the nDCG scores observed for each group; however, we also present the standard deviation of them (σ) in parentheses to analyze the ranking accuracy performances more comprehensively.

To analyze the proposed approaches' trade-off performance between ranking accuracy and popularity bias, we also conducted one-tailed t-tests for examining whether both the improvements in ΔGAP values are statistically significant and the deteriorations in nDCG scores are not statistically significant at the 95% confidence level. In doing so, we consider two different comparisons as explained in the following: (*i*) both the *AdaptedVaR* and our *ERP* variants against the pure aggregation technique and (*ii*) our *ERP* variants against the best-performing *AdaptedVaR* method in terms of both ΔGAP and nDCG, as given in the footnotes of the tables. Note that the greater the improvements in ΔGAP values and the lower the deteriorations in nDCG scores, the more qualified the recommendation lists.

As can be seen in Tables 9, 10, and 11, the ΔGAP values obtained from experiments performed for MLP, MLM, and Ciao20 datasets demonstrate that utilizing one of the AV, AU, or Mul techniques as the aggregation method usually leads to higher ΔGAP values, i.e., higher popularity bias, in group recommendations in comparison with remaining strategies for all group sizes. The main reason for this outcome is that these techniques tend to feature the most favored items among group members while performing aggregation.

The lowest ΔGAP values, on the other hand, are observed with the techniques aimed at reducing misery, i.e., AwM and LM, for each different setting. This is because they consider only a subset of the provided ratings for an item rather than its all votes during the aggregation process; thus, the items having a relatively smaller number of ratings are considered recommendable even if they are unpopular in the group, enabling diminishing the popularity bias in the produced group recommendations. Such findings also holds true for the CAMRa2011 dataset, as can be followed by Table 12.

Based on the nDCG results presented in Tables 9, 10, 11, and 12, it can be concluded that the BC outperforms all other aggregation techniques in ranking accuracy. Also, all aggregation techniques and their modified versions usually achieve higher nDCG results on the MLM dataset compared to others for each different setting. This observation's main reason is that the average number of ratings that an item received in the MLM dataset (around 253 ratings) outnumbers those in the others, making the utilized aggregation technique more efficient as more ratings are incorporated into the aggregation process. Besides, the empirical outcomes suggest that group size has no significant effect on the aggregation techniques' nDCG performances. On the other hand, the ΔGAP values obtained by the AV, AU, Mul, and BC usually decrease as the group size is getting smaller, while those achieved by other techniques increase. Note that since user groups in the CAMRa2011 dataset are pre-defined, we cannot analyze the effect of group size on the group recommendation performance for this dataset.

The outcomes of the experiments conducted for all datasets also demonstrate that modifying aggregation techniques by the *AdaptedVaR* method usually decreases ΔGAP values, which

means it improves recommendation quality by reducing popularity bias in produced group recommendations. This observation becomes more apparent in the $\alpha = 0.75$ setting than in lower configurations since higher α values in the *AdaptedVaR* method causes penalizing popular items more and picking up long-tail items in group recommendations.

For all datasets, the popularity-bias improvements achieved by the *AdaptedVaR* ($\alpha = 0.75$) usually seem to be statistically significant at the %95 confidence level for all aggregation techniques, except for LM and AwM. This finding is more evident in small user groups than the large ones. Moreover, this optimal setting of the *AdaptedVaR* causes a negligible loss with the *n*DCG scores for all aggregation techniques, except for some settings of AU, AV, Mul, and BC. Thus, it can be concluded that incorporating aggregation techniques along with the *AdaptedVaR* method is usually beneficial in producing group recommendations less-biased against popularity.

As an overall evaluation of the experimental outcomes performed for four different datasets, it can be concluded that empowering aggregation techniques with either of the proposed ERP_{Aug} and ERP_{Mul} procedures significantly decrease popularity-bias in recommendations and enhance recommendation quality in almost all schemes. Also, both strategies usually outperform the best-performing *AdaptedVaR* variant, i.e., *AdaptedVaR* ($\alpha = 0.75$), in terms of ΔGAP values, which highlights the both proposed schemes as a better alternative strategy.

In comparing two, empirical outcomes suggest that the ERP_{Mul} usually outperforms the ERP_{Aug} in counteracting popularity bias in group recommendations. The main reason of a such finding is that the ERP_{Aug} employs the weights reflecting the popularity of items inversely as an additive factor in estimating ranking scores, while the ERP_{Mul} utilizes them as a factoring coefficient, which leads to more penalizing favored items in produced recommendations. However, when the user groups are small, utilizing the ERP_{Mul} leads to a dramatic deterioration in the *n*DCG scores of almost all aggregation techniques. On the other hand, the ERP_{Aug} does not lead to a significant decrease in the *n*DCG scores and even improves ranking accuracy for some particular schemes of the AU, AV, and MP techniques. Furthermore, the standard deviations (σ) in the *n*DCG values calculated for groups usually seem to be similar, except for the ERP_{Mul} approach, which are even dramatically higher than the other settings. This indicates that the ERP_{Mul} leads to having recommendations that are not in similar quality for all group sizes. Therefore, the overall insight gained through all experiments demonstrates that ERP_{Aug} is more robust than the ERP_{Mul} considering the trade-off between conflicting goals of obtaining higher ranking accuracy and lower popularity-bias.

In conclusion, the conducted experiments in both datasets suggest that utilizing the proposed ERP_{Aug} debiasing procedure leads to substantial gains in group recommendation quality in terms of popularity-bias, without any considerable losses in ranking accuracy.

		$\Delta GAP \Downarrow$			nDCG $\Uparrow$ ($\sigma \Downarrow$)		
Group Size (P)		Small (P = 10)	Medium (P = 50)	Large (P = 100)	Small (P = 10)	Medium (P = 50)	Large (P = 100)
Technique	Method						
Avg	Pure-Avg	-0.41	-0.65	-0.75	0.78 (.05)	0.73 (.04)	0.72 (.04)
	AdaptedVaR (α = 0.25)	-0.27	-0.66	-0.80†	0.77 (.04)	0.73 (.05)	0.72 (.04)
	AdaptedVaR (α = 0.50)	-0.18	-0.66	-0.80†	0.76 (.04)	0.72 (.05)	0.71 (.04)
	AdaptedVaR (α = 0.75)	-0.17	-0.70†	−0.83†	0.75* (.05)	0.72 (.06)	0.71 (.05)
	ERP_{Mul}	-0.62$^{\dagger\diamond}$	-0.82$^{\dagger\diamond}$	-0.88$^{\dagger\diamond}$	0.69^{*+} (.26)	0.70* (.10)	0.70 (.04)
	ERP_{Aug}	-0.51$^{\dagger\diamond}$	-0.79$^{\dagger\diamond}$	-0.87$^{\dagger\diamond}$	0.76 (.05)	0.72 (.04)	0.71 (.04)
AV	Pure-AV	0.51	0.84	0.91	0.79 (.06)	0.81 (.03)	0.81 (.03)
	AdaptedVaR (α = 0.25)	0.51	0.81†	0.90†	0.80 (.04)	0.81 (.03)	0.81 (.03)
	AdaptedVaR (α = 0.50)	0.08†	-0.03†	-0.06†	0.78 (.04)	0.76* (.04)	0.76* (.03)
	AdaptedVaR (α = 0.75)	-0.17†	-0.66†	-0.80†	0.76* (.05)	0.70* (.06)	0.70* (.05)
	ERP_{Mul}	-0.16†	0.11†	0.18†	0.69^{*+} (.27)	0.79 (.11)	0.82 (.03)
	ERP_{Aug}	0.43†	0.74†	0.78†	0.80 (.04)	0.82 (.03)	0.82 (.02)
AU	Pure-AU	0.63	0.93	0.97	0.80 (.04)	0.80 (.03)	0.80 (.02)
	AdaptedVaR (α = 0.25)	0.62	0.92	0.97	0.80 (.04)	0.80 (.03)	0.80 (.02)
	AdaptedVaR (α = 0.50)	0.13†	0.03†	0.04†	0.79 (.04)	0.77 (.04)	0.77 (.03)
	AdaptedVaR (α = 0.75)	-0.16†	-0.65†	-0.80†	0.78 (.05)	0.72* (.06)	0.71* (.05)
	ERP_{Mul}	-0.14†	0.13†	0.18†	0.71^{*+} (.27)	0.80 (.11)	0.80 (.03)
	ERP_{Aug}	0.55†	0.83†	0.91†	0.81 (.04)	0.81 (.03)	0.81 (.02)
Mul	Pure-Mul	0.64	0.94	0.97	0.79 (.04)	0.80 (.03)	0.80 (.02)
	AdaptedVaR (α = 0.25)	0.25†	-0.09†	-0.23†	0.79 (.04)	0.73* (.07)	0.71* (.06)
	AdaptedVaR (α = 0.50)	0.12†	-0.20†	-0.33†	0.79 (.04)	0.72* (.07)	0.70* (.06)
	AdaptedVaR (α = 0.75)	-0.16†	-0.65†	-0.80†	0.78 (.05)	0.68* (.09)	0.65* (.07)
	ERP_{Mul}	0.13†	0.57†	0.68†	0.70^{*+} (.27)	0.78 (.10)	0.80 (.03)
	ERP_{Aug}	0.43†	0.73†	0.77†	0.80 (.04)	0.80 (.03)	0.80 (.02)
LM	Pure-LM	-0.52	-0.74	-0.84	0.74 (.05)	0.71 (.05)	0.71 (.06)
	AdaptedVaR (α = 0.25)	-0.25	-0.68	-0.82	0.73 (.05)	0.70 (.05)	0.69 (.04)
	AdaptedVaR (α = 0.50)	-0.19	-0.66	-0.80	0.73 (.05)	0.70 (.06)	0.68* (.05)
	AdaptedVaR (α = 0.75)	-0.17	-0.65	-0.80	0.73 (.05)	0.70 (.06)	0.68* (.05)
	ERP_{Mul}	-0.62$^{\dagger\diamond}$	-0.84$^{\dagger\diamond}$	-0.90$^{\dagger\diamond}$	0.61^{*+} (.24)	0.68* (.10)	0.69 (.05)
	ERP_{Aug}	-0.52$^{\diamond}$	-0.82$^{\dagger\diamond}$	-0.90$^{\dagger\diamond}$	0.74 (.05)	0.71 (.05)	0.70 (.04)
MP	Pure-MP	-0.09	-0.20	-0.33	0.74 (.04)	0.71 (.04)	0.70 (.04)
	AdaptedVaR (α = 0.25)	-0.21†	-0.67†	-0.81†	0.74 (.05)	0.71 (.05)	0.70 (.05)
	AdaptedVaR (α = 0.50)	-0.17†	-0.66†	-0.80†	0.74 (.05)	0.70 (.06)	0.70 (.05)
	AdaptedVaR (α = 0.75)	-0.17†	-0.65†	-0.80†	0.73 (.05)	0.70 (.06)	0.70 (.05)
	ERP_{Mul}	-0.61$^{\dagger\diamond}$	-0.82$^{\dagger\diamond}$	-0.88$^{\dagger\diamond}$	0.66^{*+} (.26)	0.69 (.10)	0.70 (.04)
	ERP_{Aug}	-0.50$^{\dagger\diamond}$	-0.79$^{\dagger\diamond}$	-0.88$^{\dagger\diamond}$	0.75 (.04)	0.72 (.04)	0.71 (.03)
AwM	Pure-AwM	-0.53	-0.78	-0.83	0.72 (.07)	0.71 (.06)	0.73 (.04)
	AdaptedVaR (α = 0.25)	-0.29	-0.70	-0.80	0.72 (.06)	0.73 (.05)	0.73 (.04)
	AdaptedVaR (α = 0.50)	-0.24	-0.67	-0.78	0.72 (.06)	0.73 (.05)	0.73 (.04)
	AdaptedVaR (α = 0.75)	-0.24	-0.67	-0.78	0.70 (.06)	0.73 (.05)	0.73 (.04)
	ERP_{Mul}	-0.62$^{\dagger\diamond}$	-0.81$^{\dagger\diamond}$	-0.87$^{\dagger\diamond}$	0.64^{*+} (.28)	0.69 (.08)	0.71 (.09)
	ERP_{Aug}	-0.50$^{\dagger\diamond}$	-0.79$^{\dagger\diamond}$	-0.84$^{\dagger\diamond}$	0.72 (.05)	0.70 (.04)	0.73 (.03)
BC	Pure-BC	0.45	0.76	0.81	0.81 (.05)	0.82 (.04)	0.82 (.04)
	AdaptedVaR (α = 0.25)	0.40†	0.72†	0.81	0.81 (.05)	0.82 (.05)	0.82 (.04)
	AdaptedVaR (α = 0.50)	-0.02†	-0.13†	-0.15†	0.78 (.06)	0.75* (.05)	0.76* (.05)
	AdaptedVaR (α = 0.75)	-0.16†	-0.65†	-0.80†	0.76* (.06)	0.69* (.06)	0.69* (.06)
	ERP_{Mul}	-0.30$^{\dagger\diamond}$	-0.05†	0.04†	0.69^{*+} (.27)	0.77* (.11)	0.80 (.04)
	ERP_{Aug}	-0.22$^{\dagger\diamond}$	-0.12†	-0.08†	0.81 (.04)	0.82 (.03)	0.82 (.03)

$\Downarrow$ and $\Uparrow$; desired trend for the indicator

† For significance at 95%; popularity-bias improvement w.r.t. pure technique

$\diamond$ For significance at 95%; popularity-bias improvement w.r.t. *AdaptedVaR* variant having the smallest ΔGAP

$*$ For significance at 95%; ranking accuracy deterioration w.r.t. pure technique

+ For significance at 95%; ranking accuracy deterioration w.r.t. *AdaptedVaR* variant having the highest nDCG

Table 9: ΔGAP and nDCG results for the MLP dataset

		$\Delta GAP \Downarrow$			nDCG $\Uparrow$ ($\sigma \Downarrow$)		
Group Size (P)		Small ($P = 10$)	Medium ($P = 50$)	Large ($P = 100$)	Small ($P = 10$)	Medium ($P = 50$)	Large ($P = 100$)
Technique	Method						
Avg	Pure-Avg	-0.33	-0.61	-0.75	0.80 (.05)	0.84 (.05)	0.85 (.05)
	AdaptedVaR ($\alpha = 0.25$)	-0.34	-0.60	-0.74	0.81 (.05)	0.81 (.05)	0.84 (.05)
	AdaptedVaR ($\alpha = 0.50$)	-0.36†	-0.64†	-0.80†	0.82 (.05)	0.83 (.05)	0.84 (.05)
	AdaptedVaR ($\alpha = 0.75$)	-0.40†	-0.68†	-0.83†	0.81 (.05)	0.82 (.05)	0.83 (.05)
	ERP_{Mul}	-0.58$^{\dagger\diamond}$	-0.79$^{\dagger\diamond}$	-0.87$^{\dagger\diamond}$	0.75^{*+} (.25)	0.81 (.10)	0.83 (.07)
	ERP_{Aug}	-0.48$^{\dagger\diamond}$	-0.78$^{\dagger\diamond}$	-0.86$^{\dagger\diamond}$	0.80 (.05)	0.81 (.04)	0.83 (.04)
AV	Pure-AV	0.73	1.18	1.30	0.89 (.07)	0.91 (.04)	0.92 (.04)
	AdaptedVaR ($\alpha = 0.25$)	0.72	1.16	1.28	0.89 (.07)	0.91 (.04)	0.92 (.0t)
	AdaptedVaR ($\alpha = 0.50$)	0.21†	0.21†	0.23†	0.86 (.05)	0.86* (.04)	0.78* (.05)
	AdaptedVaR ($\alpha = 0.75$)	-0.09†	-0.59†	-0.75†	0.84* (.06)	0.80* (.05)	0.79* (.05)
	ERP_{Mul}	-0.18$^{\dagger\diamond}$	-0.08†	-0.01†	0.78^{*+} (.26)	0.88 (.11)	0.89 (.07)
	ERP_{Aug}	0.12†	0.34†	0.41†	0.90 (.06)	0.92 (.04)	0.92 (.04)
AU	Pure-AU	0.83	1.25	1.36	0.89 (.05)	0.89 (.04)	0.89 (.04)
	AdaptedVaR ($\alpha = 0.25$)	0.81	1.24	1.35	0.90 (.05)	0.91 (.04)	0.91 (.04)
	AdaptedVaR ($\alpha = 0.50$)	0.24†	0.26†	0.27†	0.89 (.05)	0.88 (.04)	0.88 (.04)
	AdaptedVaR ($\alpha = 0.75$)	-0.09†	-0.58†	-0.74†	0.87 (.05)	0.83* (.05)	0.81* (.05)
	ERP_{Mul}	-0.03$^{\dagger\diamond}$	0.29†	0.34†	0.80^{*+} (.27)	0.89 (.10)	0.88 (.06)
	ERP_{Aug}	0.73†	1.16†	1.28†	0.91 (.04)	0.92 (.04)	0.90 (.03)
Mul	Pure-Mul	0.83	1.26	1.36	0.87 (.05)	0.86 (.04)	0.84 (.04)
	AdaptedVaR ($\alpha = 0.25$)	0.43†	0.06†	-0.10†	0.89 (.06)	0.83 (.07)	0.79* (.08)
	AdaptedVaR ($\alpha = 0.50$)	0.25†	-0.08†	-0.20†	0.88 (.06)	0.82* (.07)	0.78* (.08)
	AdaptedVaR ($\alpha = 0.75$)	-0.09†	-0.58†	-0.74†	0.81* (.07)	0.77* (.10)	0.72* (.11)
	ERP_{Mul}	0.43†	1.00†	1.15†	0.87 (.27)	0.89 (.10)	0.90 (.06)
	ERP_{Aug}	0.53†	0.85†	0.93†	0.90 (.05)	0.91 (.04)	0.91 (.04)
LM	Pure-LM	-0.39	-0.71	-0.84	0.80 (.06)	0.79 (.05)	0.78 (.05)
	AdaptedVaR ($\alpha = 0.25$)	-0.24	-0.70	-0.85	0.79 (.06)	0.77 (.05)	0.77 (.05)
	AdaptedVaR ($\alpha = 0.50$)	-0.21	-0.69	-0.85	0.79 (.06)	0.77 (.06)	0.77 (.06)
	AdaptedVaR ($\alpha = 0.75$)	-0.19	-0.69	-0.85	0.78 (.06)	0.76 (.06)	0.75 (.06)
	ERP_{Mul}	-0.68$^{\dagger\diamond}$	-0.90$^{\dagger\diamond}$	-0.98$^{\dagger\diamond}$	0.69^{*+} (.24)	0.73* (.09)	0.73* (.07)
	ERP_{Aug}	-0.58$^{\dagger\diamond}$	-0.88$^{\dagger\diamond}$	-0.97$^{\dagger\diamond}$	0.77 (.05)	0.74 (.05)	0.73* (.05)
MP	Pure-MP	-0.06	-0.16	-0.21	0.84 (.05)	0.80 (.05)	0.78 (.04)
	AdaptedVaR ($\alpha = 0.25$)	-0.12†	-0.60†	-0.75†	0.85 (.05)	0.80 (.05)	0.78 (.05)
	AdaptedVaR ($\alpha = 0.50$)	-0.09†	-0.59†	-0.75†	0.84 (.05)	0.80 (.05)	0.78 (.05)
	AdaptedVaR ($\alpha = 0.75$)	-0.09†	-0.59†	-0.75†	0.84 (.05)	0.80 (.05)	0.78 (.05)
	ERP_{Mul}	-0.58$^{\dagger\diamond}$	-0.80$^{\dagger\diamond}$	-0.87$^{\dagger\diamond}$	0.75^{*+} (.25)	0.78 (.08)	0.78 (.06)
	ERP_{Aug}	-0.48$^{\dagger\diamond}$	-0.78$^{\dagger\diamond}$	-0.87$^{\dagger\diamond}$	0.83 (.05)	0.80 (.04)	0.78 (.04)
AwM	Pure-AwM	-0.60	-0.74	-0.85	0.85 (.06)	0.81 (.05)	0.80 (.05)
	AdaptedVaR ($\alpha = 0.25$)	-0.35	-0.63	-0.78	0.85 (.06)	0.81 (.05)	0.80 (.05)
	AdaptedVaR ($\alpha = 0.50$)	-0.32	-0.61	-0.77	0.85 (.06)	0.81 (.05)	0.80 (.05)
	AdaptedVaR ($\alpha = 0.75$)	-0.32	-0.61	-0.77	0.84 (.05)	0.80 (.05)	0.80 (.05)
	ERP_{Mul}	-0.68$^{\dagger\diamond}$	-1.00$^{\dagger\diamond}$	-0.91$^{\dagger\diamond}$	0.75^{*+} (.25)	0.79 (.08)	0.78 (.06)
	ERP_{Aug}	-0.49$^{\diamond}$	-0.99$^{\dagger\diamond}$	-0.90$^{\dagger\diamond}$	0.84 (.05)	0.80 (.05)	0.79 (.05)
BC	Pure-BC	0.64	1.11	1.26	0.91 (.05)	0.92 (.05)	0.91 (.04)
	AdaptedVaR ($\alpha = 0.25$)	0.55†	1.06†	1.23	0.90 (.05)	0.91 (.05)	0.91 (.04)
	AdaptedVaR ($\alpha = 0.50$)	-0.01†	-0.11†	-0.01†	0.86* (.05)	0.83* (.05)	0.84* (.05)
	AdaptedVaR ($\alpha = 0.75$)	-0.09†	-0.19†	-0.25†	0.85* (.06)	0.79* (.06)	0.76* (.06)
	ERP_{Mul}	-0.26$^{\dagger\diamond}$	-0.32$^{\dagger\diamond}$	-0.36$^{\dagger\diamond}$	0.77^{*+} (.26)	0.85^{*+} (.10)	0.86^{*+} (.07)
	ERP_{Aug}	-0.13$^{\dagger\diamond}$	-0.16†	-0.20†	0.89 (.05)	0.91 (.04)	0.91 (.04)

$\Downarrow$ and $\Uparrow$; desired trend for the indicator

† For significance at 95%; popularity-bias improvement w.r.t. pure technique

$\diamond$ For significance at 95%; popularity-bias improvement w.r.t. *AdaptedVaR* variant having the smallest ΔGAP

$*$ For significance at 95%; ranking accuracy deterioration w.r.t. pure technique

$+$ For significance at 95%; ranking accuracy deterioration w.r.t. *AdaptedVaR* variant having the highest nDCG

Table 10: ΔGAP and nDCG results for the MLM dataset

		$\Delta GAP \Downarrow$			nDCG $\Uparrow$ ($\sigma \Downarrow$)		
Group Size (P)		Small (P = 10)	Medium (P = 50)	Large (P = 100)	Small (P = 10)	Medium (P = 50)	Large (P = 100)
Technique	Method						
Avg	Pure-Avg	0.25	-0.05	-0.21	0.77 (.11)	0.79 (.06)	0.80 (.04)
	AdaptedVaR (α = 0.25)	0.21†	-0.07†	-0.33†	0.77 (.11)	0.79 (.06)	0.80 (.04)
	AdaptedVaR (α = 0.50)	0.12†	-0.11†	-0.31†	0.76 (.11)	0.79 (.06)	0.80 (.05)
	AdaptedVaR (α = 0.75)	-0.03†	-0.15†	-0.30†	0.75 (.11)	0.78 (.07)	0.79 (.05)
	ERP_{Mul}	-0.39$^{\dagger\diamond}$	-0.54$^{\dagger\diamond}$	-0.64$^{\dagger\diamond}$	0.52^{*+} (.40)	0.72^{*+} (.22)	0.79 (.12)
	ERP_{Aug}	-0.09$^{\dagger\diamond}$	-0.46$^{\dagger\diamond}$	-0.62$^{\dagger\diamond}$	0.77 (.11)	0.79 (.06)	0.80 (.03)
AV	Pure-AV	0.34	1.06	1.42	0.75 (.14)	0.77 (.08)	0.78 (.05)
	AdaptedVaR (α = 0.25)	0.41	1.02†	1.38†	0.75 (.14)	0.77 (.08)	0.78 (.06)
	AdaptedVaR (α = 0.50)	0.36	0.36†	0.37†	0.74 (.11)	0.77 (.07)	0.78 (.05)
	AdaptedVaR (α = 0.75)	0.30	-0.04†	-0.28†	0.73 (.11)	0.76 (.06)	0.77 (.05)
	ERP_{Mul}	-0.27$^{\dagger\diamond}$	0.25†	0.48†	0.51^{*+} (.40)	0.72^{*+} (.24)	0.67^{*+} (.12)
	ERP_{aUG}	0.12†	0.26†	0.32†	0.75 (.11)	0.77 (.06)	0.78 (.05)
AU	Pure-AU	0.45	1.20	1.58	0.75 (.11)	0.76 (.06)	0.76 (.04)
	AdaptedVaR (α = 0.25)	0.45	1.14	1.52	0.75 (.11)	0.76 (.06)	0.76 (.04)
	AdaptedVaR (α = 0.50)	0.40†	0.42†	0.43†	0.75 (.11)	0.74 (.06)	0.75 (.05)
	AdaptedVaR (α = 0.75)	0.32†	0.03†	-0.07†	0.74 (.12)	0.73* (.07)	0.72* (.06)
	ERP_{Mul}	-0.25$^{\dagger\diamond}$	-0.13$^{\dagger\diamond}$	-0.15$^{\dagger\diamond}$	0.62^{*+} (.41)	0.72* (.24)	0.77 (.12)
	ERP_{Aug}	0.35†	0.95†	1.02†	0.76 (.11)	0.77 (.05)	0.77 (.04)
Mul	Pure-Mul	0.45	1.19	1.56	0.75 (.11)	0.76 (.06)	0.76 (.04)
	AdaptedVaR (α = 0.25)	0.42	0.53†	0.45†	0.75 (.11)	0.76 (.06)	0.76 (.04)
	AdaptedVaR (α = 0.50)	0.38†	0.34†	0.36†	0.75 (.11)	0.75 (.06)	0.76 (.04)
	AdaptedVaR (α = 0.75)	0.33†	0.12†	0.04†	0.74 (.12)	0.75 (.07)	0.74 (.05)
	ERP_{Mul}	-0.16$^{\dagger\diamond}$	-0.03$^{\dagger\diamond}$	-0.27$^{\dagger\diamond}$	0.62^{*+} (.36)	0.70^{*+} (.23)	0.75 (.12)
	ERP_{Aug}	0.39†	0.98†	0.85†	0.75 (.11)	0.76 (.05)	0.76 (.04)
LM	Pure-LM	-0.07	-0.27	-0.37	0.70 (.17)	0.74 (.09)	0.76 (.05)
	AdaptedVaR (α = 0.25)	0.04	-0.17	-0.37	0.70 (.17)	0.74 (.10)	0.76 (.06)
	AdaptedVaR (α = 0.50)	0.10	-0.12	-0.34	0.69 (.17)	0.74 (.10)	0.76 (.06)
	AdaptedVaR (α = 0.75)	0.15	-0.09	-0.32	0.68 (.17)	0.73 (.11)	0.75 (.07)
	ERP_{Mul}	-0.40$^{\dagger\diamond}$	-0.54$^{\dagger\diamond}$	-0.64$^{\dagger\diamond}$	0.58^{*+} (.37)	0.67^{*+} (.23)	0.71^{*+} (.12)
	ERP_{Aug}	-0.16$^{\dagger\diamond}$	-0.47$^{\dagger\diamond}$	-0.62$^{\dagger\diamond}$	0.68 (.12)	0.74 (.06)	0.75 (.04)
MP	Pure-MP	0.26	-0.17	-0.30	0.76 (.11)	0.78 (.06)	0.79 (.04)
	AdaptedVaR (α = 0.25)	0.25	-0.08†	-0.32†	0.76 (.11)	0.78 (.06)	0.78 (.04)
	AdaptedVaR (α = 0.50)	0.22†	-0.06†	-0.31†	0.75 (.11)	0.77 (.06)	0.78 (.04)
	AdaptedVaR (α = 0.75)	0.17†	-0.05†	-0.30†	0.74 (.11)	0.77 (.07)	0.78 (.05)
	ERP_{Mul}	-0.39$^{\dagger\diamond}$	-0.54$^{\dagger\diamond}$	-0.64$^{\dagger\diamond}$	0.62^{*+} (.41)	0.72^{*+} (.24)	0.77 (.12)
	ERP_{Aug}	-0.09$^{\dagger\diamond}$	-0.46$^{\dagger\diamond}$	-0.62$^{\dagger\diamond}$	0.77 (.10)	0.79 (.05)	0.80 (.03)
AwM	Pure-AwM	-0.14	-0.35	-0.48	0.75 (.14)	0.79 (.06)	0.80 (.04)
	AdaptedVaR (α = 0.25)	-0.18†	-0.12	-0.33	0.75 (.14)	0.79 (.06)	0.80 (.04)
	AdaptedVaR (α = 0.50)	-0.21†	-0.10	-0.33	0.75 (.15)	0.79 (.07)	0.79 (.04)
	AdaptedVaR (α = 0.75)	-0.23†	-0.10	-0.32	0.74 (.15)	0.77 (.07)	0.77 (.05)
	ERP_{Mul}	-0.47$^{\dagger\diamond}$	-0.53$^{\dagger\diamond}$	-0.64$^{\dagger\diamond}$	0.63^{*+} (.41)	0.73^{*+} (.23)	0.77 (.12)
	ERP_{Aug}	-0.41$^{\dagger\diamond}$	-0.46$^{\dagger\diamond}$	-0.62$^{\dagger\diamond}$	0.75 (.13)	0.79 (.05)	0.79 (.04)
BC	Pure-BC	0.31	0.92	1.31	0.79 (.11)	0.80 (.06)	0.79 (.04)
	AdaptedVaR (α = 0.25)	0.32	0.75†	1.11†	0.76 (.11)	0.77 (.06)	0.78 (.04)
	AdaptedVaR (α = 0.50)	0.32	0.23†	0.13†	0.75 (.11)	0.78 (.06)	0.79 (.04)
	AdaptedVaR (α = 0.75)	0.28†	-0.04†	-0.09†	0.74 (.12)	0.77 (.06)	0.79 (.05)
	ERP_{Mul}	-0.33$^{\dagger\diamond}$	-0.17$^{\dagger\diamond}$	-0.12$^{\dagger\diamond}$	0.62^{*+} (.41)	0.73^{*+} (.24)	0.77 (.13)
	ERP_{Aug}	-0.19$^{\dagger\diamond}$	-0.01†	-0.05†	0.76 (.11)	0.77 (.05)	0.79 (.04)

$\Downarrow$ and $\Uparrow$; desired trend for the indicator

$\dagger$ For significance at 95%; popularity-bias improvement w.r.t. pure technique

$\diamond$ For significance at 95%; popularity-bias improvement w.r.t. *AdaptedVaR* variant having the smallest ΔGAP

$*$ For significance at 95%; ranking accuracy deterioration w.r.t. pure technique

$+$ For significance at 95%; ranking accuracy deterioration w.r.t. *AdaptedVaR* variant having the highest nDCG

Table 11: ΔGAP and nDCG results for the Ciao20 dataset

Technique	Method	$\Delta GAP \Downarrow$	nDCG $\Uparrow$ ($\sigma \Downarrow$)	Technique	Method	$\Delta GAP \Downarrow$	nDCG $\Uparrow$ ($\sigma \Downarrow$)
Avg	Pure-Avg	0.48	0.81 (.06)	LM	Pure-LM	-0.22	0.80 (.06)
	AdaptedVaR ($\alpha = 0.25$)	0.47	0.81 (.06)		*AdaptedVaR* ($\alpha = 0.25$)	0.14	0.80 (.06)
	AdaptedVaR ($\alpha = 0.50$)	$0.27^{\dagger}$	0.79 (.06)		*AdaptedVaR* ($\alpha = 0.50$)	0.46	0.80 (.06)
	AdaptedVaR ($\alpha = 0.75$)	$0.18^{\dagger}$	0.76^{*} (.06)		*AdaptedVaR* ($\alpha = 0.75$)	0.48	0.75^{*} (.06)
	ERP_{Mul}	$-0.26^{\dagger\diamond}$	0.75^{*+} (.06)		ERP_{Mul}	$-0.28^{\dagger\diamond}$	0.76^{*} (.06)
	ERP_{Aug}	$-0.08^{\dagger\diamond}$	0.81 (.06)		ERP_{Aug}	$-0.26^{\dagger\diamond}$	0.79 (.06)
AV	Pure-AV	0.51	0.71 (.05)	MP	Pure-MP	0.03	0.76 (.06)
	AdaptedVaR ($\alpha = 0.25$)	0.51	0.71 (.05)		*AdaptedVaR* ($\alpha = 0.25$)	0.44	0.76 (.07)
	AdaptedVaR ($\alpha = 0.50$)	$0.47^{\dagger}$	0.71 (.05)		*AdaptedVaR* ($\alpha = 0.50$)	0.48	0.76 (.07)
	AdaptedVaR ($\alpha = 0.75$)	$0.47^{\dagger}$	0.71 (.05)		*AdaptedVaR* ($\alpha = 0.75$)	0.48	0.75 (.07)
	ERP_{Mul}	$-0.25^{\dagger\diamond}$	0.64^{*+} (.06)		ERP_{Mul}	$-0.26^{\dagger\diamond}$	0.75 (.06)
	ERP_{Aug}	$0.46^{\dagger}$	0.71 (.05)		ERP_{Aug}	$-0.26^{\dagger\diamond}$	0.76 (.06)
AU	Pure-AU	0.52	0.81 (.06)	AwM	Pure-AwM	-0.19	0.79 (.06)
	AdaptedVaR ($\alpha = 0.25$)	0.52	0.79 (.06)		*AdaptedVaR* ($\alpha = 0.25$)	0.31	0.79 (.06)
	AdaptedVaR ($\alpha = 0.50$)	$0.48^{\dagger}$	0.80 (.06)		*AdaptedVaR* ($\alpha = 0.50$)	0.47	0.78 (.06)
	AdaptedVaR ($\alpha = 0.75$)	$0.48^{\dagger}$	0.77^{*} (.06)		*AdaptedVaR* ($\alpha = 0.75$)	0.47	0.76^{*} (.06)
	ERP_{Mul}	$-0.25^{\dagger\diamond}$	0.77^{*} (.06)		ERP_{Mul}	$-0.26^{\dagger\diamond}$	0.77 (.06)
	ERP_{Aug}	$0.46^{\dagger}$	0.80 (.05)		ERP_{Aug}	$-0.26^{\dagger\diamond}$	0.78 (.06)
Mul	Pure-Mul	0.52	0.81 (.06)	BC	Pure-BC	0.33	0.83 (.06)
	AdaptedVaR ($\alpha = 0.25$)	0.52	0.81 (.07)		*AdaptedVaR* ($\alpha = 0.25$)	0.31	0.82 (.07)
	AdaptedVaR ($\alpha = 0.50$)	0.52	0.79 (.06)		*AdaptedVaR* ($\alpha = 0.50$)	$0.25^{\dagger}$	0.82 (.06)
	AdaptedVaR ($\alpha = 0.75$)	$0.48^{\dagger}$	0.77^{*} (.06)		*AdaptedVaR* ($\alpha = 0.75$)	$0.19^{\dagger}$	0.81 (.06)
	ERP_{Mul}	$-0.22^{\dagger\diamond}$	0.76^{*+} (.06)		ERP_{Mul}	$-0.05^{\dagger\diamond}$	0.75^{*+} (.06)
	ERP_{Aug}	$0.47^{\dagger}$	0.81 (.06)		ERP_{Aug}	$0.03^{\dagger\diamond}$	0.82 (.05)

$\Downarrow$ and $\Uparrow$; desired trend for the indicator

$\dagger$ For significance at 95%; popularity-bias improvement w.r.t. pure technique

$\diamond$ For significance at 95%; popularity-bias improvement w.r.t. *AdaptedVaR* variant having the smallest ΔGAP

$*$ For significance at 95%; ranking accuracy deterioration w.r.t. pure technique

$+$ For significance at 95%; ranking accuracy deterioration w.r.t. *AdaptedVaR* variant having the highest nDCG

Table 12: ΔGAP and nDCG results for the CAMRa2011 dataset

6. Conclusions and future work

Investigating bias of the recommendation algorithms against popular items and treating its adverse effects play a vital role in producing high-quality ranking-based recommendations. Although some practical solutions have been introduced for personalized recommendations, the issues of popularity bias are not considered previously in group recommender systems, which are enhanced mechanisms aiming to produce appropriate referrals for groups of users rather than individuals.

The aggregation techniques are the vital components of such systems as group recommendations are commonly generated based on the group ratings estimated by these techniques. Therefore, in this study, we first profoundly analyze eight conventional aggregation techniques (i.e., *average*, *multiplicative*, *additive utilitarian*, *average without misery*, *approval voting*, *least misery*, *most pleasure*, and *borda count*), in terms of popularity bias they induce in group recommendation scenarios. The experiments performed on two benchmark datasets demonstrate that the techniques focusing on reducing misery, i.e., *average without misery* and *least misery*, are the most successful in penalizing popular items and picking up the long-tail ones compared to other aggregation techniques. On the other hand, the *additive utilitarian, approval voting* and *multiplicative* aggregation techniques lead to by far the highest popularity bias in group recommendations due to their nature featuring the most favored items among group members in aggregating.

We adopt a classic popularity bias mitigation approach developed for personalized recom-

mendations named *Value-aware Ranking (VaR)* and adapt it to group recommendation scenarios. More specifically, the adapted version of the *(VaR)* approach, namely the *AdaptedVaR*, initially calculates a weight for each item inversely proportional to their popularity in the group. It then determines ranking scores for items by calculating the weighted average of the estimated weights and the produced group ratings using a proper aggregation technique. Such ranking scores are latterly used to re-ranking items in providing top-N group recommendations. Empirical outcomes suggest that incorporating aggregation techniques with the *AdaptedVaR* usually decreases popularity-bias in the produced recommendations, with negligible losses in ranking accuracy.

We also propose the *Enhanced Re-ranking Procedure (ERP)* that helps to counteract popularity bias during the aggregation process and improve group recommendation quality. This procedure reaches ultimate ranking scores for items by following two different combining strategies, termed as *Multiplicative (*ERP_{Mul}*)* and *Augmentative (*ERP_{Aug}*)*. Specifically, the former focuses on highly penalizing popular items in estimating ultimate ranking scores by utilizing the calculated item weights as a factoring coefficient on the estimated group ratings. On the other hand, the latter values group ratings as the decisive factor in ultimate ranking scores and employs the item weights in a different role, as the auxiliary influencer. The empirical outcomes indicate that both strategies are more efficient than the *AdaptedVaR*, and empowering aggregation techniques with either of the ERP_{Mul} and ERP_{Aug} significantly decreases biases towards popular items. The obtained results also demonstrate that the ERP_{Aug} is more robust than the ERP_{Mul} considering the trade-off between ranking accuracy and popularity bias.

Group recommendation approaches are beneficial tools to overcome some context or cost problems that may appear in individual recommendations. Therefore, the popular digital platforms, such as Netflix or Steam, can efficiently employ our proposed *ERP* methods, especially the ERP_{Aug}, to achieve more qualified recommendation lists where the popularity-bias issue is treated. Thus, they can provide more visibility of the niche items in their produced recommendation lists and boost such items' sales. However, the main shortcoming of the proposed ERP variants is that they cannot be directly applied to some particular aggregation techniques aimed at ranking priority, such as *plurality voting*. This is because such aggregation mechanisms focus on producing a ranked list of items rather than generating group ratings for items.

Although the effectiveness of our proposed popularity-debiasing methods is verified in terms of ranking accuracy, future research might include evaluating these methods with beyond-accuracy metrics such as fairness and coverage. Also, since each member's interest level in a group towards popular items might differ, the ERP procedures can be improved by integrating such different propensities of members in the ranking score estimation process.

References

Abdollahpouri, H., Burke, R., Mobasher, B., 2017. Controlling popularity bias in learning-to-rank recommendation, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, Association for Computing Machinery, New York, NY, USA. p. 42–46. doi:https://doi.org/10.1145/3109859.3109912.

Abdollahpouri, H., Burke, R., Mobasher, B., 2018. Popularity-aware item weighting for long-tail recommendation. arXiv:1802.05382.

Abdollahpouri, H., Burke, R., Mobasher, B., 2019a. Managing popularity bias in recommender systems with personalized re-ranking. arXiv:1901.07555.

Abdollahpouri, H., Mansoury, M., Burke, R., Mobasher, B., 2019b. The impact of popularity bias on fairness and calibration in recommendation. arXiv:1910.05755.

Abdollahpouri, H., Mansoury, M., Burke, R., Mobasher, B., 2019c. The unfairness of popularity bias in recommendation. arXiv:1907.13286.

Agarwal, A., Chakraborty, M., Chowdary, C.R., 2017. Does order matter? effect of order in group recommendation. Expert Systems with Applications 82, 115–127. doi:https://doi.org/10.1016/j.eswa.2017.03.069.

Ahmad, H.S., Nurjanah, D., Rismala, R., 2017. A combination of individual model on memory-based group recommender system to the books domain, in: 2017 5th International Conference on Information and Communication Technology (ICoIC7), pp. 1–6. doi:https://doi.org/10.1109/ICoICT.2017.8074655.

Ardissono, L., Goy, A., Petrone, G., Segnan, M., Torasso, P., 2003. Intrigue: Personalized recommendation of tourist attractions for desktop and hand held devices. Applied Artificial Intelligence 17, 687–714. doi:https://doi.org/10.1080/713827254.

Baltrunas, L., Makcinskas, T., Ricci, F., 2010. Group recommendations with rank aggregation and collaborative filtering, in: Proceedings of the Fourth ACM Conference on Recommender Systems, Association for Computing Machinery, New York, NY, USA. p. 119–126. doi:https://doi.org/10.1145/1864708.1864733.

Barzegar Nozari, R., Koohi, H., 2020. A novel group recommender system based on members' influence and leader impact. Knowledge-Based Systems 205, 106296. doi:https://doi.org/10.1016/j.knosys.2020.106296.

Bawden, D., Robinson, L., 2020. Information overload: An introduction, in: Oxford Research Encyclopedia of Politics.

Bobadilla, J., Serradilla, F., Bernal, J., 2010. A new collaborative filtering metric that improves the behavior of recommender systems. Knowledge-Based Systems 23, 520 – 528. doi:https://doi.org/10.1016/j.knosys.2010.03.009.

Boratto, L., Carta, S., 2014. Using collaborative filtering to overcome the curse of dimensionality when clustering users in a group recommender system, in: Proceedings of the 16th International Conference on Enterprise Information Systems - Volume 2, SCITEPRESS - Science and Technology Publications, Lda, Setubal, PRT. p. 564–572. doi:https://doi.org/10.5220/0004865005640572.

Boratto, L., Carta, S., 2015. Art: Group recommendation approaches for automatically detected groups. International Journal of Machine Learning and Cybernetics 6, 953–980. doi:https://doi.org/10.1007/s13042-015-0371-4.

Boratto, L., Carta, S., Fenu, G., 2016. Discovery and representation of the preferences of automatically detected groups: Exploiting the link between group modeling and clustering. Future Generation Computer Systems 64, 165 – 174. doi:https://doi.org/10.1016/j.future.2015.10.007.

Boratto, L., Carta, S., Satta, M., 2010. Groups identification and individual recommendations in group recommendation algorithms., in: PRSAT@ recsys, pp. 27–34.

Boratto, L., Fenu, G., Marras, M., 2019. The effect of algorithmic bias on recommender systems for massive open online courses, in: Azzopardi, L., Stein, B., Fuhr, N., Mayr, P., Hauff, C., Hiemstra, D. (Eds.), Advances in Information Retrieval, Springer International Publishing, Cham. pp. 457–472. doi:https://doi.org/10.1007/978-3-030-15712-8_30.

Boratto, L., Fenu, G., Marras, M., 2021. Connecting user and item perspectives in popularity debiasing for collaborative recommendation. Information Processing & Management 58, 102387. doi:https://doi.org/10.1016/j.ipm.2020.102387.

Cañamares, R., Castells, P., 2018. Should i follow the crowd? a probabilistic analysis of the effectiveness of popularity in recommender systems, Association for Computing Machinery, New York, NY, USA. p. 415–424. doi:https://doi.org/10.1145/3209978.3210014.

Castro, J., Yera, R., Martínez, L., 2017. An empirical study of natural noise management in group recommendation systems. Decision Support Systems 94, 1–11. doi:https://doi.org/10.1016/j.dss.2016.09.020.

Castro, J., Yera, R., Martínez, L., 2018. A fuzzy approach for natural noise management in group recommender systems. Expert Systems with Applications 94, 237 – 249. doi:https://doi.org/10.1016/j.eswa.2017.10.060.

Chao, D.L., Balthrop, J., Forrest, S., 2005. Adaptive radio: Achieving consensus using negative preferences, in: Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work, ACM, New York, NY, USA. pp. 120–123. doi:https://doi.org/10.1145/1099203.1099224.

Chen, C., Zhang, M., Liu, Y., Ma, S., 2018. Missing data modeling with user activity and item popularity in recommendation, in: Tseng, Y.H., Sakai, T., Jiang, J., Ku, L.W., Park, D.H., Yeh, J.F., Yu, L.C., Lee, L.H., Chen, Z.H. (Eds.), Information Retrieval Technology, Springer International Publishing, Cham. pp. 113–125. doi:10.1007/978-3-030-03520-4_11.

Christensen, I., Schiaffino, S., Armentano, M., 2016. Social group recommendation in the tourism domain. Journal of Intelligent Information Systems 47, 209–231. doi:https://doi.org/10.1007/s10844-016-0400-0.

Christensen, I.A., Schiaffino, S., 2011. Entertainment recommender systems for group of users. Expert Systems with Applications 38, 14127 – 14135. doi:https://doi.org/10.1016/j.eswa.2011.04.221.

Crossen, A., Budzik, J., Hammond, K.J., 2002. Flytrap: Intelligent group music recommendation, in: Proceedings of the 7th International Conference on Intelligent User Interfaces, ACM, New York, NY, USA. pp. 184–185. doi:https://doi.org/10.1145/502716.502748.

Felfernig, A., Boratto, L., Stettinger, M., Tkalčič, M., 2018. Evaluating Group Recommender Systems. Springer International Publishing, Cham. pp. 59–71. doi:https://doi.org/10.1007/978-3-319-75067-5_3.

Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J., 1999. An algorithmic framework for performing collaborative filtering, in: Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY, USA. pp. 230–237. doi:https://doi.org/10.1145/312624.312682.

Hou, L., Pan, X., Liu, K., 2018. Balancing the popularity bias of object similarities for personalised recommendation. The European Physical Journal B 91, 1–7. doi:https://doi.org/10.1140/epjb/e2018-80374-8.

Jameson, A., 2004. More than the sum of its members: Challenges for group recommender systems, in: Proceedings of the Working Conference on Advanced Visual Interfaces, Association for Computing Machinery, New York, NY, USA. p. 48–54. doi:https://doi.org/10.1145/989863.989869.

Jannach, D., Kamehkhosh, I., Bonnin, G., 2016. Biases in automated music playlist generation: A comparison of next-track recommending techniques, Association for Computing Machinery, New York, NY, USA. p. 281–285. doi:https://doi.org/10.1145/2930238.2930283.

Jannach, D., Lerche, L., Kamehkhosh, I., Jugovac, M., 2015. What recommenders recommend: An analysis of recommendation biases and possible countermeasures. User Modeling and User-Adapted Interaction 25, 427–491. doi:10.1007/s11257-015-9165-3.

Kamishima, T., Akaho, S., Asoh, H., Sakuma, J., 2014. Correcting popularity bias by enhancing recommendation neutrality, in: RecSys Posters.

Kim, H.N., El Saddik, A., 2015. A stochastic approach to group recommendations in social media systems. Information Systems 50, 76–93. doi:https://doi.org/10.1016/j.is.2014.10.002.

Koren, Y., 2008. Factorization meets the neighborhood: A multifaceted collaborative filtering model, Association for Computing Machinery, New York, NY, USA. p. 426–434. doi:https://doi.org/10.1145/1401890.1401944.

Kowald, D., Schedl, M., Lex, E., 2020. The unfairness of popularity bias in music recommendation: A reproducibility study, in: Jose, J.M., Yilmaz, E., Magalhães, J., Castells, P., Ferro, N., Silva, M.J., Martins, F. (Eds.), Advances in Information Retrieval, Springer International Publishing, Cham. pp. 35–42. doi:https://doi.org/10.1007/978-3-030-45442-5_5.

Masthoff, J., 2015. Group Recommender Systems: Aggregation, Satisfaction and Group Attributes. Springer US, Boston, MA. pp. 743–776. doi:https://doi.org/10.1007/978-1-4899-7637-6_22.

McCarthy, J.E., Anagnost, T.D., 2000. Musicfx: An arbiter of group preferences for computer supported collaborative workouts, in: Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, Association for Computing Machinery, New York, NY, USA. p. 348. doi:https://doi.org/10.1145/358916.361976.

McCarthy, J.F., 2002. Pocket restaurantfinder: A situated recommender system for groups, in: Workshop on Mobile Ad-Hoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems.

McCarthy, K., Salamó, M., Coyle, L., McGinty, L., Smyth, B., Nixon, P., 2006. Cats: A synchronous approach to collaborative group recommendation, in: FLAIRS 2006 - Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference, AAAI Press. pp. 86–91.

Nguyen, T.N., Ricci, F., 2017. A chat-based group recommender system for tourism, in: Information and communication technologies in tourism 2017. Springer, pp. 17–30. doi:https://doi.org/10.1007/s40558-017-0099-y.

O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J., 2001. PolyLens: A Recommender System for Groups of Users. Springer Netherlands, Dordrecht. pp. 199–218. doi:https://doi.org/10.1007/0-306-48019-0_11.

Oh, J., Park, S., Yu, H., Song, M., Park, S., 2011. Novel recommendation based on personal popularity tendency, in: 2011 IEEE 11th International Conference on Data Mining, pp. 507–516.

Park, Y.J., Tuzhilin, A., 2008. The long tail of recommender systems and how to leverage it, Association for Computing Machinery, New York, NY, USA. p. 11–18. doi:10.1145/1454008.1454012.

Quijano-Sanchez, L., Recio-Garcia, J.A., Diaz-Agudo, B., 2011. Happymovie: A facebook application for recommending movies to groups, in: 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence, pp. 239–244. doi:https://doi.org/10.1109/ICTAI.2011.44.

Sacharidis, D., 2019. Top-n group recommendations with fairness, in: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, Association for Computing Machinery, New York, NY, USA. p. 1663–1670. doi:https://doi.org/10.1145/3297280.3297442.

Sánchez, P., 2019. Exploiting contextual information for recommender systems oriented to tourism, Association for Computing Machinery, New York, NY, USA. p. 601–605. URL: https://doi.org/10.1145/3298689.3347062, doi:10.1145/3298689.3347062.

Seo, Y.D., Kim, Y.G., Lee, E., Seol, K.S., Baik, D.K., 2018. An enhanced aggregation method considering deviations for a group recommendation. Expert Systems with Applications 93, 299 – 312. doi:https://doi.org/10.1016/j.eswa.2017.10.027.

Shambour, Q., 2021. A deep learning based algorithm for multi-criteria recommender systems. Knowledge-Based Systems 211, 106545. doi:https://doi.org/10.1016/j.knosys.2020.106545.

Van Lierop, D., Badami, M.G., El-Geneidy, A.M., 2018. What influences satisfaction and loyalty in public transport? a review of the literature. Transport Reviews 38, 52–72. doi:https://doi.org/10.1080/01441647.2017.

1298683.

Vozalis, M., Margaritis, K.G., 2004. Collaborative filtering enhanced by demographic correlation, in: AIAI symposium on professional practice in AI, of the 18th world computer congress.

Vozalis, M., Margaritis, K.G., 2006. On the enhancement of collaborative filtering by demographic data. Web Intelligence and Agent Systems: An International Journal 4, 117–138.

Wang, X., Liu, Y., Lu, J., Xiong, F., Zhang, G., 2019. Trugrc: Trust-aware group recommendation with virtual coordinators. Future Generation Computer Systems 94, 224 – 236. doi:https://doi.org/10.1016/j.future.2018.11.030.

Yalcin, E., Bilge, A., 2020. A personality-based aggregation technique for group recommendation. Eskişehir Technical University Journal of Science and Technology A-Applied Sciences and Engineering 21, 486–498. doi:https://doi.org/10.18038/estubtda.743422.

Yalcin, E., Bilge, A., 2021. Novel automatic group identification approaches for group recommendation. Expert Systems with Applications , 114709doi:https://doi.org/10.1016/j.eswa.2021.114709.

Yalcin, E., Ismailoglu, F., Bilge, A., 2021. An entropy empowered hybridized aggregation technique for group recommender systems. Expert Systems with Applications 166, 114111. doi:https://doi.org/10.1016/j.eswa.2020.114111.

Zhiwen, Y., Xingshe, Z., Daqing, Z., 2005. An adaptive in-vehicle multimedia recommender for group users, in: 2005 IEEE 61st Vehicular technology conference, IEEE. pp. 2800–2804. doi:https://doi.org/10.1109/vetecs.2005.1543857.